DOCUMENT RESUME

ED 125 660 IR 003 752

TITLE	Computer Assisted Instruction. Papers Presented at the Association for Educational Data Systems Annual Convention (Phoenix, Arizona, May 3-7, 1976).
INSTITUTION	Association for Educational Data Systems, Washington, D.C.
PUB DATE	May 76
NOTE	120p.; For related documents, see IR 003 748-756; Some parts may be marginally legible due to print quality of original
AVAILABLE FROM	Association for Educational Data Systems, 1201 Sixteenth Street, N.W., Washington, D.C. 20036 ($10.00 for entire proceedings)
EDRS PRICE	MF-$0.83 HC-$6.01 Plus Postage.
DESCRIPTORS	Autoinstructional Aids; Case Studies; *Computer Assisted Instruction; Computer Graphics; *Computer Programs; *Computer Science Education; Educational Innovation; Elementary Education; Higher Education; Instructional Media; Instructional Technology; Man Machine Systems; Mathematics Education; *On Line Systems; Research Utilization; Statistical Analysis; Technological Advancement
IDENTIFIERS	AEDS 76; *Association for Educational Data Systems; Interactive Computer Systems

ABSTRACT

Two abstracts and seventeen articles on computer assisted instruction (CAI) presented at the 1976 Association for Educational Data Systems (AEDS) convention are included here. Four new computer programs are described: Author System for Education and Training (ASET); GNOSIS, a Swedish/English CAI package; Statistical Interactive Programming System (SIPS); and Instructional Dialogue Facility (IDF) which helps classroom teachers learn CAI languages. Six papers examine interactive computer systems and interactive CAI, and one paper describes an interactive educational system which uses video discs in conjunction with computers. Six papers describe various programs: a CAI course in logic instruction for department of defense personnel, CAI for computer science education, computers for drill and practice in math, and general use of CAI in elementary school classrooms. A computer program which monitors student progress in calculus is described, and user groups are the topic of one article. (CH)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED125660

COMPUTER ASSISTED INSTRUCTION. Papers presented at the Association for Educational Data Systems Annual Convention.

BEST COPY AVAILABLE

Alfred Bork
John Robson
Donna Hutchenson
Dennis H. Sorge
C. Edward Streeter
Richard F. Conklin
Robert L. DeMichiell
Frank Greenwood
Richard G. Montanelli
D. A. Michalopoulos
Edwin L. Anderson
Susan Finch
Shirley Chann
Clare Hamlet
L. P. Huelsman
Lotus M. Knief
John Luiten
Julia Feugate
Paul Furnas
Sally Yeates Sedelow
E. O. G Davies
Mary Cooper
Gary R. Tobey
Major Robert H. Orr
Jutith A. Watson
William R. Johnson

IR 003 752

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

ASSOCIATION FOR EDUCATIONAL DATA SYSTEMS

The 1976 AEDS International Convention

Proceedings

TODAY'S REVOLUTION:
COMPUTERS IN EDUCATION

May 3 - 7, 1976

Phoenix, Arizona

These Proceedings have been made possible by a grant from
the IBM Corporation as a service to the educational community.

COMPUTER DIALOGS AT IRVINE - ONE MONTH'S ACTIVITY

Alfred Bork
Physics Computer Development Project
University of California
Irvine, California 92717

The Physics Computer Development Project at the University of California, Irvine, is involved in the development and use of computer-based teaching materials, primarily in the sciences. Several recent review papers about our activities have been published.[1] These papers typically consider the Project objectives, the types of dialogs available, and the authoring procedures for developing dialogs.

The present paper gives a different cross section on how computers are used in education at Irvine by concentrating on a single month's activity, January, 1976. It considers both dialog usage and dialog developmental work during that period. Naturally any one month will have peculiarities associated with it, but this mode of presentation may be of some interest in viewing an ongoing project that is developing computer-based dialogs. Many other class uses, such as those involving student programming, are not considered.

Irvine Computer Facilities

The University of California, Irvine, is one of the three new campuses of the nine-campus University of California system, established about ten years ago. Ralph Gerard, our first dean of graduate studies, and Daniel G. Aldrich, Jr., our Chancellor, became convinced at that time that computers were to play an increasingly important role in the learning process. Hence, Irvine acquired more facilities for using computers with students, than is typical on university campuses. After some rough beginnings we finally arrived at a stable configuration of computers about seven years ago.

The campus now operates three general purpose computers in competition with each other. The computers are a Sigma 7 (the one used in all the activities to be reported in this paper), a PDP-10, and a PDP-11/45. All three are available for all types of work, and the two larger systems, the Sigma 7 and the PDP-10, support both batch and timesharing. The Sigma 7 has about seventy physical ports and the PDP-10 has about thirty physical ports. Although we have no accurate figures of the numbers of terminals on campus, it is estimated that perhaps 175 terminals are on campus, about 100 generally accessible to students.

The work of the Physics Computer Development Project, funded by the National Science Foundation and the University of California, is almost entirely dependent on graphic capabilities, so graphic terminals are employed in most of the activities reported in this discussion. We currently have about thirty Tektronix 4013 terminals available for student use, about twenty-five on the Sigma 7. Although some of the dialog activities can take place on nongraphic terminals, most require graphics.

January Classes

Most of the student-computer dialogs developed in the Physics Computer Development Project have been intended for beginning science classes. Which materials are heavily used in a particular month is dependent on which classes are being taught, and where those classes are in their syllabi. Most of the use described took place in three classes. One was the Math 1 class, a preparation for beginning calculus. Students take Math 1 only if they fail an entrance exam in the calculus course. A second important class was the Physics 3B class, the second quarter of a three-quarter general physics survey taken primarily by biology majors, mostly premedical students. The third course, Physics 5A, is the first quarter of a five-quarter introductory course primarily for science and engineering majors, also with a good sprinkling of biology students. In the Physics 3B course the computer plays a major role in the structure of the course. In the Physics 5A course computer activity is simply another learning mode among others available for use if the student desires.

It should not be implied that all the use of the computer at this time took place in just these classes. Considerable additional access comes from students not enrolled in any physics classes employing the dialogs. This nonclass use tends to spread much more throughout the entire range of dialogs. (There are perhaps seventy dialogs for student use available in all areas at Irvine at the present time.) So this "free" use of dialogs does not show up in the major dialog usage to be reported here.

January Usage

Computer dialogs keep in files accurate usage figures, so we know just which dialogs are used, by whom, and how much. About once a month we sort this file; and the following data were obtained from that source.

Dialogs were used during January, 1976 for approximately 120,000 terminal minutes. About 900 individuals, almost all students, were involved in about 8,000 separate sessions. These figures for January are typical of those in recent months during the current academic year.

It should be indicated that there is not completely "free" use at Irvine. The three classes mentioned share a group of sixteen "ports." If a student in one of the classes tries to sign on when all sixteen ports are busy, he or she will receive a message to that effect and will not be able to gain access to the computer. The availability of terminals in the busy times of day might also be another limiting factor; there are three concentrations of graphic terminals, one between the computers themselves, one on the first floor of the Physical Science building, and one on the fourth floor of the Physical Science building, and these terminals are heavily employed. Although Irvine provides more student access to computing than many universities, computing is still a valuable commodity, and we do not have a completely free access market. It is difficult to predict what usage would be without constraints, but it would be considerably larger.

The following table shows the statistics for the most widely used dialogs at Irvine.

SELECTED DIALOG USAGE - JANUARY, 1976

	Terminal Minutes	Students	Sessions	Minutes/ Student	Minutes/ Session
CHARGE	3,785	139	297	27	13
CHEM	4,736	26	430	182	11
DIALOGS	1,103	141	693	8	2
FIELD	31,622	258	1,266	123	25
GLEM	644	53	117	12	6
GRAPH	1,268	43	167	29	8
HOCKEY	602	34	49	18	12
INTEGRAL	7,711	137	271	56	28
LUNA	1,217	49	108	24	11
MACQ	548	9	15	61	36
MOTION	1,649	61	125	27	13
NEIL	768	48	131	16	6
QZ1	18,950	249	639	76	30
QZ10	874	18	44	48	20
QZ2	5,188	102	214	51	24
QZ3	998	42	73	24	14
QZ7	4,380	84	238	52	18
QZ8	2,123	55	137	39	26
QZ9	2,807	28	81	100	35
RAYS	586	18	81	33	7

In the left-hand columns are the names of the dialogs. The reader should remember that this does not represent total dialog usage, but only the dialogs used most heavily, more than 500 minutes of terminal time during the month. The heading "Students" is slightly misleading. While most of this is student use, some developmental use, to be described below, is also represented; about twenty of the "students" were actually developers of the dialog material.

In the next few sections I will review particular dialogs, describing them and telling a little bit about the class environment in which they were used.

The Math Quizzes

As indicated above, Math 1 is a "remedial" math class taken by students expected to have difficulty with the calculus courses. It was a large class, with approximately 300 students in the winter quarter. The class is taught in the PSI or Keller plan strategy, with students moving at their own individual paces through the units. All the exams, needed to show success in each unit, are taken directly on-line to the computer. These exams are picked out of a large pool of questions, so that students get a different exam each time they take the quiz. The student attempts the quiz as many times as is needed to demonstrate that the material has been mastered.

On the chart the math quizzes start with the letters "QZ." Thus about seven different quizzes were used in January, indicating that the students were indeed dispersed in different parts of the course, typical in a Keller plan course. The quizzes have a built-in timing limit; the student is not allowed to spend more than one hour in taking a single quiz. You can see that they average about twenty-five minutes per quiz. Statistics also indicate that for most of the quizzes students typically take them about twice before succeeding.

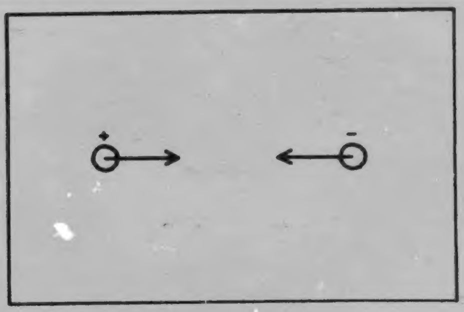
The Math 1 quizzes were developed by John Grover and Stephen Franklin of the Math Department at UCI. Comparisons of their use versus other imilarly versatile ways of teaching the course indicate that this mode of teaching is cost effective.[2]

FIELD

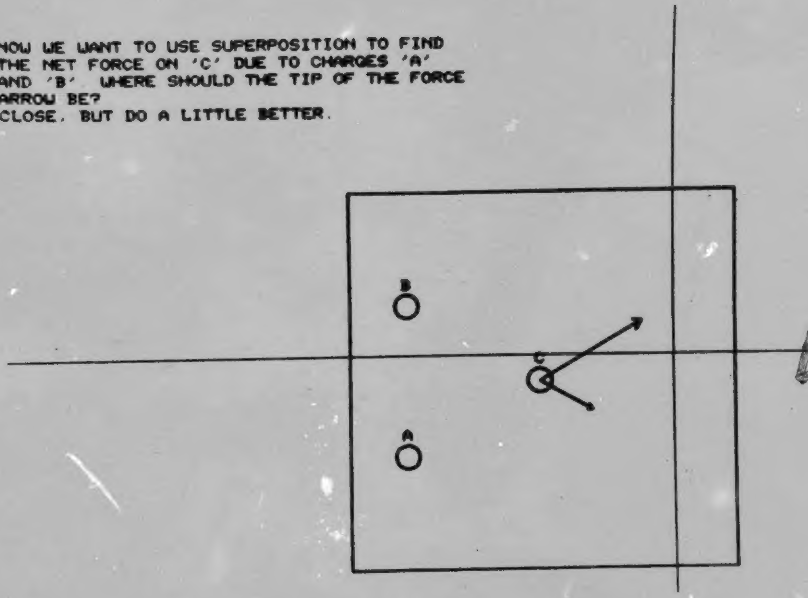
About one-quarter of the total use of Irvine dialogs during the month of January came from a single dialog, FIELD. The subject matter of FIELD is electrostatics. It carries the student from very simple beginning, considering the direction of the forces between charged particles, into the sophisticated notions of the electric field and field lines. The student plays an active role in the learning process. FIELD ends with a field-plotting facility in which students specify arbitrary locations for charges, and then ask for either electric field lines or equipotential lines. The dialog has a map at the beginning, allowing students to jump into any particular section. They can return to this map at some later time, and then move to a different section of the program.

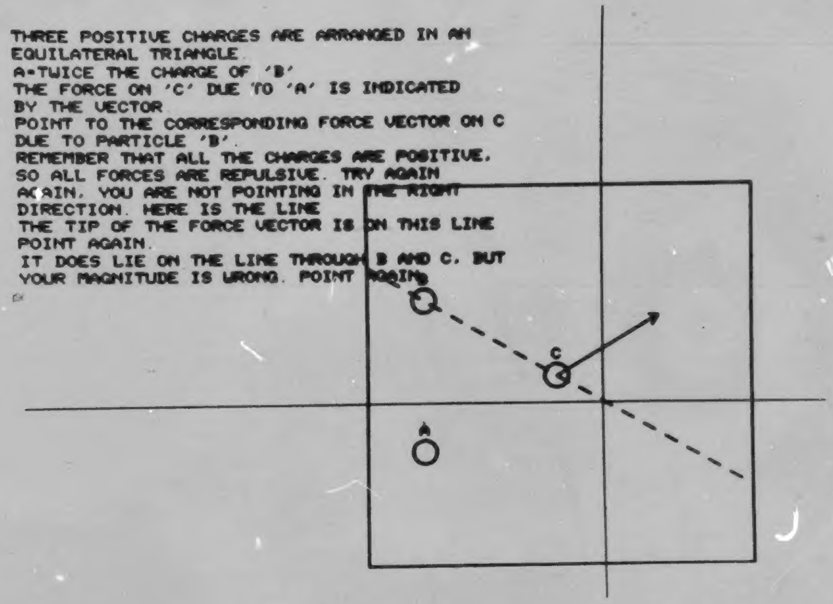
The large use of FIELD represents the Physics 3B course. The second quarter of that course is concerned with electricity and magnetism, and starts with electrostatics. FIELD was recommended to students as a viable alternate to

```
NOW WE HAVE TWO UNLIKE CHARGES
(SUCH AS AN ELECTRON AND A PROTON).
DO THEY ATTRACT OR REPEL?
?ATTRACT?

GOOD
UNLIKE CHARGES ATTRACT

PRESS RETURN TO CONTINUE...■
```

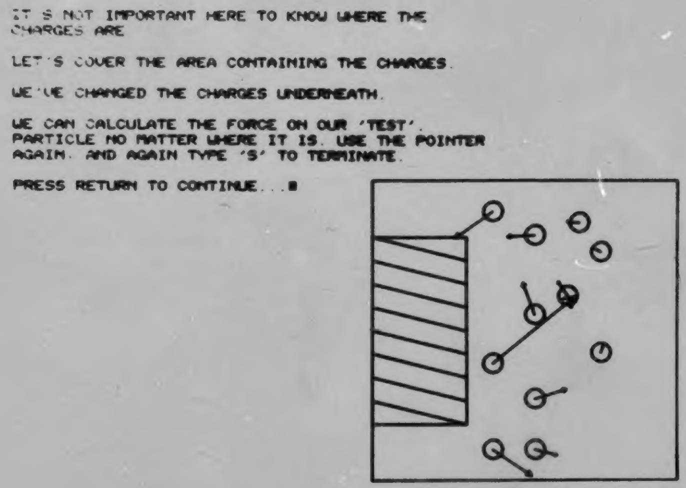


```
NOW WE WANT TO USE SUPERPOSITION TO FIND
THE NET FORCE ON 'C' DUE TO CHARGES 'A'
AND 'B'. WHERE SHOULD THE TIP OF THE FORCE
ARROW BE?
CLOSE, BUT DO A LITTLE BETTER.
```

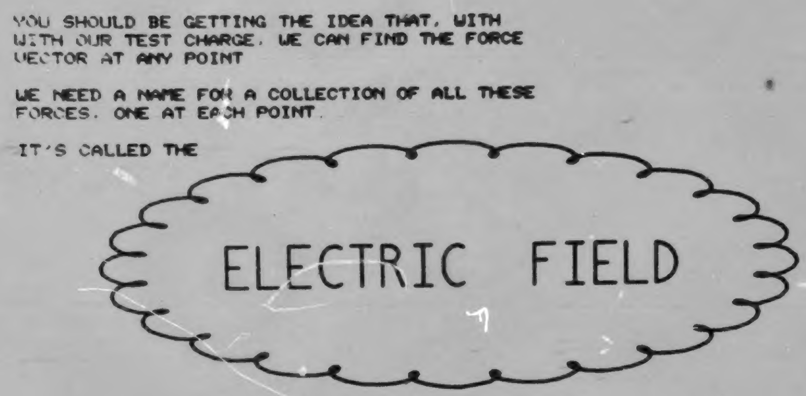




```
THREE POSITIVE CHARGES ARE ARRANGED IN AN
EQUILATERAL TRIANGLE.
A=TWICE THE CHARGE OF 'B'
THE FORCE ON 'C' DUE TO 'A' IS INDICATED
BY THE VECTOR
POINT TO THE CORRESPONDING FORCE VECTOR ON C
DUE TO PARTICLE 'B'.
REMEMBER THAT ALL THE CHARGES ARE POSITIVE,
SO ALL FORCES ARE REPULSIVE. TRY AGAIN
AGAIN, YOU ARE NOT POINTING IN THE RIGHT
DIRECTION. HERE IS THE LINE
THE TIP OF THE FORCE VECTOR IS ON THIS LINE
POINT AGAIN.
 IT DOES LIE ON THE LINE THROUGH B AND C. BUT
 YOUR MAGNITUDE IS WRONG. POINT AGAIN
```



```
IT'S NOT IMPORTANT HERE TO KNOW WHERE THE
CHARGES ARE

LET'S COVER THE AREA CONTAINING THE CHARGES.

WE'VE CHANGED THE CHARGES UNDERNEATH.

WE CAN CALCULATE THE FORCE ON OUR 'TEST'.
PARTICLE NO MATTER WHERE IT IS. USE THE POINTER
AGAIN. AND AGAIN TYPE 'S' TO TERMINATE.

PRESS RETURN TO CONTINUE...█
```



```
YOU SHOULD BE GETTING THE IDEA THAT, WITH
WITH OUR TEST CHARGE, WE CAN FIND THE FORCE
VECTOR AT ANY POINT

WE NEED A NAME FOR A COLLECTION OF ALL THESE
FORCES, ONE AT EACH POINT.

IT'S CALLED THE
```

learning electrostatics, in addition to the lecture and text material. Clearly many students employed it in this fashion. Also several Physics 3B problems were structured around the field plotting capabilities in the last section of FIELD, to be sure that the student had experience with the plotting of field lines; from this experience they can gain intuitive understanding for electric fields due to discrete charges.

FIELD was begun by a group of University of California physics faculty members on several campuses in a workshop sponsored by the University to make physics faculty acquainted with the dialog materials and design approach. Those involved included Sun-yiu Fung (Riverside), Robert Eisberg (Santa Barbara), Peter Geissert (Davis), Bruce Rosenblum (Santa Cruz), Alfred Bork (Irvine), Richard Ballard (Irvine). Later additions to the dialog were made by Alfred Bork and Joseph Marasco of Irvine. The student programmer responsible for FIELD was Thomas Marrs.

INTEGRAL

INTEGRAL was also a recommended dialog in the Physics 3B course, but for a different reason than for FIELD. One of the problems in teaching introductory electricity and magnetism is that current textbooks tend to make mathematical demands which go beyond those the student has seen in mathematics courses. Although Physics 3B, as with similar courses elsewhere, assumes that the students have had calculus, they will in general not be familiar with the notions of line integral and surface integral needed for a full exposition of electromagnetic theory. INTEGRAL attempts to provide this necessary mathematical background, and eventually concludes with a discussion of Gauss's law, one of the fundamental laws of electromagnetic theory.

INTEGRAL too is driven from a map, and students can use sections of it necessary to meet individual needs. Viewing the usage of INTEGRAL and FIELD it can be seen that only about half the students in the class used INTEGRAL while almost everyone used FIELD. Furthermore, the average time in INTEGRAL was only about half the average time in FIELD.

INTEGRAL was developed by Bruce Rosenblum of the University of California, Santa Cruz, in connection with the workshop mentioned above.

DIALOGS

The statistics on DIALOGS might appear puzzling until one understands what is involved. Although it was used for almost 700 sessions in January, more than any other dialog except FIELD, the average use of DIALOGS for a single session was only for two minutes. Thus, it is employed often, but briefly.

DIALOGS is an information source, telling the student which dialogs are available, and will furnish a brief description of each one. Although this same information is also available in printed form, many students will not have the printed form with them, and will use DIALOGS to find either the program they want or as a method of browsing through other dialogs. The students use it only briefly in seeking information about dialogs.

Other Comments on Student Dialog Usage

Briefer comments can be made about some of the other dialogs that played a major role in classes in January. CHARGE is an on-line problem dialog

PCDP

PHYSICS COMPUTER DEVELOPMENT PROJECT
UNIVERSITY of CALIFORNIA
at IRVINE

MAY I HELP YOU ? ■

where the student must query for information about an electrostatic situation to solve the problem. It persuades students to think about what information is necessary to determine an answer as opposed to the usual textbook situation where all necessary information is stated. It was written by Sun-Yiu Fung of the University of California, Riverside.

GLEM and NEIL are both lunar lander programs, based on simple kinematics. GLEM is graphic, while NEIL is not. Most of NEIL's use typically comes from nongraphic terminals.

Although MOTION and LUNA do not show large usage for the month of January, in a full year statistic they generally end up higher than other Irvine dialogs. MOTION was developed by Richard Ballard and Alfred Bork. It is an "F = ma" simulation, allowing the student great freedom in the choice of initial conditions, equation constants, scaling, force laws, and variables to be plotted. LUNA, a dialog prepared by Arnold Arons of the University of Washington and Alfred Bork, is directed toward understanding the phases of the moon, using this as an example of a scientific model.

GRAPH is a different type from any we have considered. It is a utility program, providing graphic capability for students who do not necessarily know much about computing. Students can type in the functions they want graphed. The dialog provides control of scaling and other aspects of graphing. This program receives less use now than it did at one time, because we now have good graphic capabilities available in APL, taught to almost all students in the beginning physics courses.

HOCKEY also follows a different approach from others shown. It was prepared by Robert Eisberg of the University of California, Santa Barbara and Peter Geissert of the University of California, Davis. It arrives at the notions of mass and momentum and momentum conservation, through computer-simulated experiments of collisions of pucks on an air table. Its use in January was primarily from Physics 5B students.

The use of MAGQ deserves some special mention. This program was used quite heavily in February, 1976, where it was a class assignment in the Physics 3B course. The students using it in January were presumably those who were somewhat ahead, making use of the fact that computer dialogs are always available, to accomplish this work before the scheduled period. Thus they offer a flexibility not available for lectures and for many other media.

Finally two of the dialogs in the list represent primarily developmental work, as opposed to student access. These two dialogs, CHEM and RAYS, will therefore be discussed in the next section.

Project Activity in January

The Physics Computer Development Project was also heavily involved in January in the continuing maintenance and development of computer-based dialogs.

The largest amount of developmental work, as seen by its presence in the dialog list above, went into the CHEM dialog. This is the first of the "new worlds" dialogs, an attempt to produce a highly interactive form of dialog where students are free to change the context of the discussion at any time. The primary developer is Richard

Ballard. February statistics reflect very heavily the initial use of this program with the introductory chemistry course.

RAYS was developed by William Parker at Irvine, and is an introduction to the subject of geometrical optics. It expects to see its first student use in the winter quarter.

Major implementation activities centered around dialogs that have been developed with two visitors in the last part of December, Arnold Arons and Richard Extermann. Arons and Alfred Bork developed four dialogs over a two-week period, three dealing with magnetic fields, and one concerning the difference between temperature and heat. Joseph Marasco and Richard Extermann developed a dialog concerning the weight of a body on the earth's surface. In dialog development in our Project the teachers work in a loose flowchart form, specifying the pedagogical decisions in complete detail but doing none of the actual coding of the material. Student programmers during January were involved in implementing these dialogs. By the time of the New York APS-AAPT meeting in early February it was possible to demonstrate the first of the magnetic field dialogs; the programmer was Martin Katz.

Other dialogs were also under development in this period. One, SLOPE, is intended for problem assistance. Another, SPACE, is an ambitious attempt to provide a detailed learning resource for relativity.

Only one entirely new dialog was prepared by the Project during the month of January. This was the work of Alfred Bork, Joseph Marasco, and John Herman. John Herman, a physics professor at Western Michigan University, is spending the winter quarter at Irvine on sabbatical from his own institution; his primary interest is converting our dialogs to run on another system, a DEC System 10. He has also participated in the Project in other ways. The dialog that we wrote is a problem assistance and testing dialog, a typical electrostatic problem in the first week of a beginning electricity and magnetism course. This dialog is as yet unimplemented.

Another major activity within the Project was the revision of the course management database, a part of our process of restructuring the beginning physics course. We had operational a database of this kind, to keep track of individual student records, for the first quarter course, taught by a PSI strategy. The second quarter is being taught in a more conventional pattern, and so it has been necessary to rework the database. Access is provided to the data for both the instructor and for individual students.

Finally a major continuing activity of the Project during recent months has been the development of an APL worksapce designed to aid the process of creating graphic aspects of dialogs. The aim is to provide a facility which would be usable by a graphic designer, in the sense of industrial design, so that dialogs could have a more interesting visual appearance. The work is primarily due to one of our student programmers, Mark Geisert. The programmer or designer can construct graphic or alphanumeric images, can move these around the screen by means of the built-in graphic pointer, can change the shape of the objects, can rotate them, can combine simple objects into composite objects, and can manipulate them in various ways. Control over text includes the ability to right justify and left justify, to center it, and to enter subscripts and superscripts. The program will itself write the computer code necessary to generate the graphic and alphanumeric images specified, and this code can be merged with that generated directly by programmers. This facility is proving to be very useful in all our recent dialog developments, including those mentioned in the present paper.

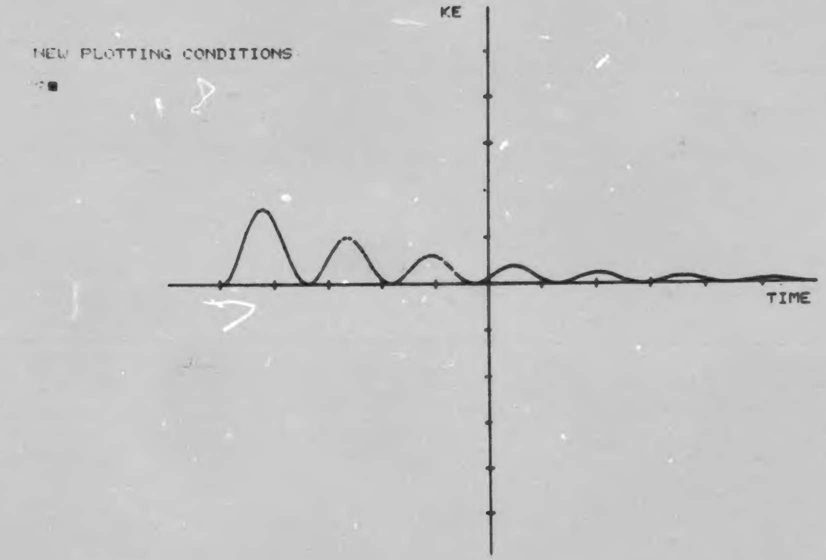
It should be stressed again that looking at any one month will give certain biases. Thus, the dialogs that received very heavy use were accidents of the particular class activities. But the general "flavor" would be the same.

The project is happy to send literature about its activities and to welcome visitors.

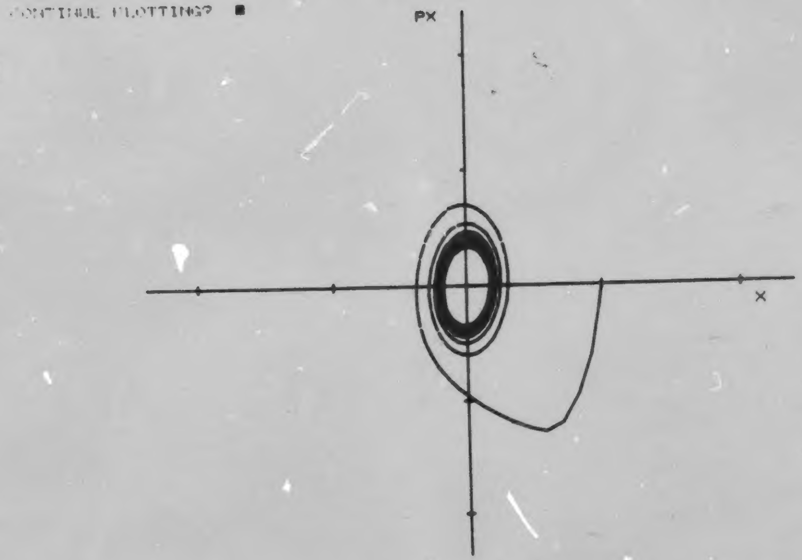
References

1. The Physics Computer Development Project, EDUCOM, Vol. 10, No. 4, Winter 1975. The Physics Computer Development Project and Student Computer Dialogs without Special Purpose Languages, Proceedings of CCUC/6, Texas Christian University, June 1975.

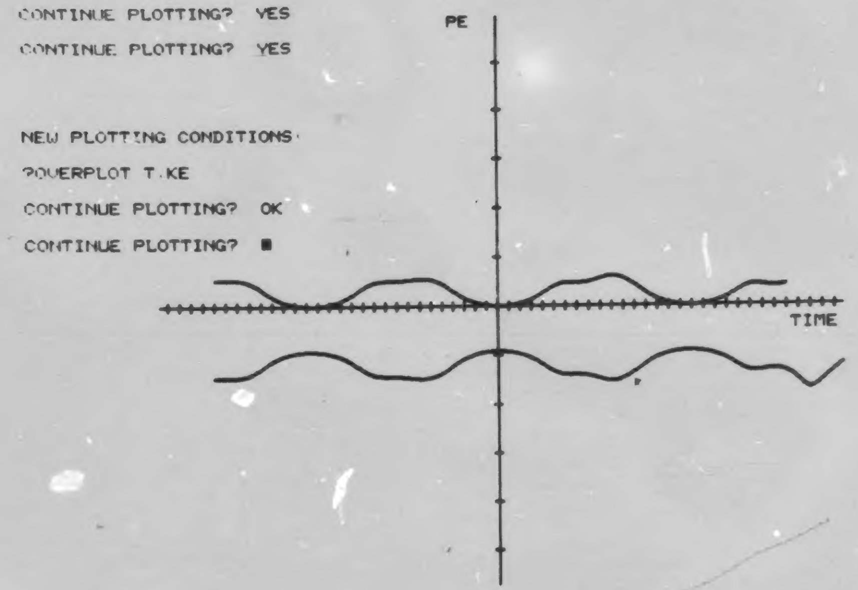
2. For additional information on the Math 1 quiz dialogs, write to Stephen Franklin, Mathematics Department, University of California, Irvine, California 92717.



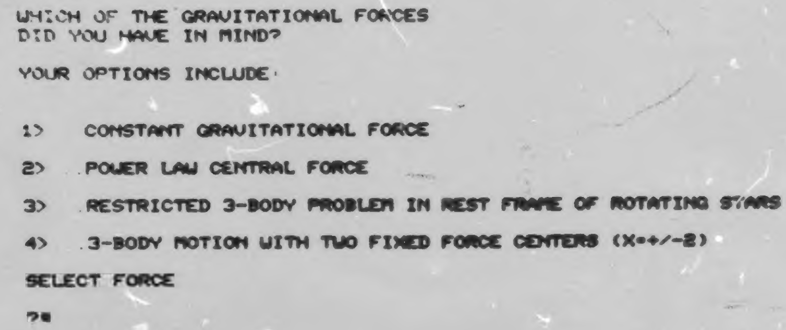
```
NEW PLOTTING CONDITIONS
?■
```



```
CONTINUE PLOTTING? ■
```



```
CONTINUE PLOTTING?  YES
CONTINUE PLOTTING?  YES

NEW PLOTTING CONDITIONS:
?OVERPLOT T,KE
CONTINUE PLOTTING?  OK
CONTINUE PLOTTING?  ■
```



```
WHICH OF THE GRAVITATIONAL FORCES
DID YOU HAVE IN MIND?

YOUR OPTIONS INCLUDE:

1>   CONSTANT GRAVITATIONAL FORCE

2>   POWER LAW CENTRAL FORCE

3>   RESTRICTED 3-BODY PROBLEM IN REST FRAME OF ROTATING STARS

4>   3-BODY MOTION WITH TWO FIXED FORCE CENTERS (X=+/-2)

SELECT FORCE

?■
```

TWO CONSUMER VIEWS OF PLATO IV

John W. Robson, Department of Physics
University of Arizona
Tucson, Arizona 85721

ABSTRACT: The Laboratory for Computer Based Instruction at the University of Arizona has operated 12 terminals connected to the PLATO IV system for approximately a year. Thus we are consumers of a service provided by the Computer-based Education Research Laboratory of the University of Illinois. Students in selected courses at the University of Arizona are consumers of instructional material delivered by PLATO. We intend to report on a year of experience from the viewpoints of these two consumer groups.

DESIGNING CAI PACKAGES FOR COMPUTER SCIENCE

Donna Hutcheson
East Texas State University
Commerce, Texas 75428

ABSTRACT: Computer Assisted Instruction (CAI) is proclaimed by many to be the cure for all educational ills. Even though that is not an accurate assumption, CAI is an excellent tool to assist instruction if adequate application packages are available. Research has shown that one of the most neglected areas of concentration for application packages is computer science. Why? Generally persons proficient in computer science are not knowledgeable in the techniques for designing an effective instructional package. To design an effective package the following steps should be employed: 1. Select a topic, 2. Define measureable objectives, 3. Write Pre and Post Tests, 4. Select instructional strategy, 5. Identify and prepare additional audio-visual materials to accompany course, 6. Code course material in an author language, 7. Functionally debug the course, 8. Conduct peer group tryout and revise, if necessary, 9. Conduct target population tryout and revise, if objectives were not met, and 10. Periodically review and maintain package.

Introduction

For the purpose of this paper, Computer Assisted Instruction (CAI) will be defined as using the computer as a tool for helping students attain the objectives for a module of instruction. Research in CAI since the inception of the concept has produced numerous articles proclaiming the excellence of CAI as an educational tool. One would think that due to the nature of CAI, computer science education would be the leader in CAI use. Yet, research conducted by Pati Smith of East Texas State University during the fall of 1975 shows that one of the most neglected areas of concentration for CAI application packages is computer science. It has been stated by some educational experts that professional educators could not write efficient CAI packages because they could not program the computer, and computer programmers could not write effective packages because they were unfamiliar with either the course material or with instructional strategies which could improve the probability for student attainment of objectives. Computer science teachers are familiar with both instructional strategies and programming techniques. So why is there not a multitude of computer science courseware available?

1. Many teachers feel that they do not have the hours of time that it takes to develop materials.
2. Most materials that have been written are not documented, and thus are useless to anyone other than the author.
3. Not all institutions use the same author language, or the same computer equipment, making exchange of materials difficult.
4. There are no clear-cut guidelines available for writing and implementing CAI packages.

This paper addresses the last reason--guidelines for developing CAI packages in the area of computer science. The following systematic approach has been successfully used at East Texas State University to train prospective computer science teachers to write effective CAI courseware.

1. Select a topic
2. Define measureable objectives
3. Write pre and post tests
4. Select instructional strategy
5. Identify and prepare additional audio-visual materials to accompany course
6. Code course material in an author language
7. Functionally debug the course
8. Conduct peer group tryout and revise, if necessary
9. Conduct target population tryout and revise, if objectives were not met
10. Periodically review and maintain package.

Steps 1-3 make up the design phase; steps 4-7, the implementation phase; and 8-10, the evaluation phase. Many beginning authors plunge directly into the implementation phase and try to write an effective course without first considering the steps involved in the design phase. Experienced CAI author/programmers generally agree that from 25 to 40 percent of the total time required to produce an efficient CAI module is spent on design. Thus, time spent on the first three steps listed above will save time and frustration in the long run.

Select a Topic

The first step, selecting a topic, is more difficult than first consideration indicates. Any teacher may be capable of designing and using instructional materials; but can the material be well adapted to CAI, and is it worth learning? Before investing the hours of work that it takes to design an effective CAI module, be sure that there is a need for the material as evidenced by students, teachers, and lack of similar materials. Also, will the segment be used once it is written?

Novice author/programmers should beware of using CAI to teach concepts which are traditionally boring even when presented by the most innovative teacher, such as history of computers. In the past, most courseware for computer science has concentrated on teaching students the syntax and semantics of a particular programming language, thus leaving the teacher free to concentrate on the refinement of programming techniques. As the author/programmer gains experience, more challenging topics should be considered.

No matter which topic is chosen, state the scope of each module or sub-module in precise terms, including the sequence in which the tasks are to be presented. Keep in mind that some tasks may not have an inherent sequence if they are prerequisites to a future task, but not prerequisite to each other (i.e. I/O devices or storage devices as prerequisites to evaluation of computer systems). Such tasks may be presented so that the student chooses the sequence of tasks to be completed.

Define Objectives

Once the topic has been selected and the scope defined, it is necessary to define measureable behavioral objectives. Authors who eliminate this step in the design phase often are frustrated later when asked "What exactly is a student supposed to learn from this course?" and "How do you know that the course taught anything?" The worth of CAI is particularly dependent upon well defined objectives in the face of traditional educators' criticism.

Write Pre and Post Tests

Now that the objectives have been identified, writing pretests and post tests is a simple matter. Many instructors and curriculum designers eliminate pretests if the content of the module is of an introductory nature and no prerequisite or entry level knowledge is necessary. However, if prerequisite knowledge is necessary for the student to succeed in any given module, a pretest is mandatory. If the student does not successfully pass the pretest, provisions for remedial materials or instructions as to what the student is to do next should be provided. Comparing the results of pretests and post tests can make learning justification arguments much easier. If the answers to the pretest questions are not given to the student at the time he makes a response, the same questions could be used in the post test. If the same questions are used, the measurement of student learning contributable directly to the CAI module is much easier. Naturally, different questions may be used in the post test. However, be sure that all questions are consistent with the previously stated objectives.

Select Instructional Strategy

At this point it will be necessary to determine the logic of the instructional strategy. The development of an effective instructional strategy requires imagination and creativity. Three factors should be considered:

1. the level of interaction between student and courseware to be utilized;
2. the teaching logic;
3. the decision algorithm to be used to select each student's path through instruction.

The level of interaction plays an important part in forming the student's reaction to CAI. Two primary levels can be identified: free-form, short-answer constructed responses, and objective-type responses (i.e. yes-no, true-false, multiple choice). At the present time most courseware contains the objective-type response format because constructed responses are more difficult to process. However, it is the opinion of this author that computer science authors should strive for constructed responses since it is only by "doing" that a student truly learns.

The teaching logic is primarily dependent on the level of interaction chosen. The following teaching logistics in order of complexity are available:

1. Drill and practice
2. Tutorial
3. Problem solving
4. Simulation
5. Inquiry

Tutorial logic is most appropriate for teaching introductory concepts in computer science, but requiring the student to apply learned concepts implies the use of either problem solving or simulation logic. It is suggested that the author/programmer try to incorporate a combination of two or three different methods in order to give the student a variety of experience, thus promoting a more complete understanding of the topic.

The decision algorithm controls the sequence of instruction. The algorithm used can range from a simple linear plan to a complex plan involving branching and numerous paths through instruction. To take full advantage of CAI's capabilities, a simple linear plan should be avoided. In the complex plan, the particular path that an individual student takes is dependent upon his performance on previous questions. Figure 1 illustrates a simple linear plan; Figure 2, a simple branching plan; and Figure 3, a complex plan. Note that the sequencing plans presented in Figures 1-3 may be used for both tutorial logical and problem solving logic. There are infinite variations of strategies between the simple linear plan and the complex plan presented. The particular strategy used should be based upon the complexity of the material being presented and the personal teaching style of the author/programmer. No matter which strategy is devised, it should be flowcharted for easier reference when coding the course into a programming language for the computer.

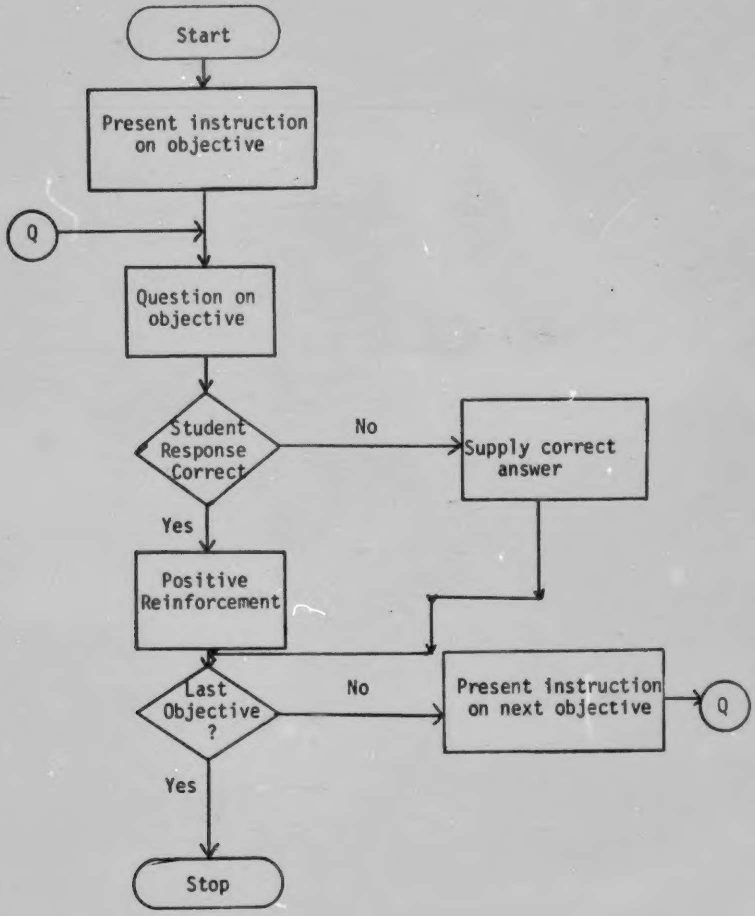

Start
Present instruction on objective
Q
Question on objective
Student Response Correct
No
Supply correct answer
Yes
Positive Reinforcement
Last Objective ?
No
Present instruction on next objective
Q
Yes
Stop

Figure 1

Instructional Strategy - Linear Sequence

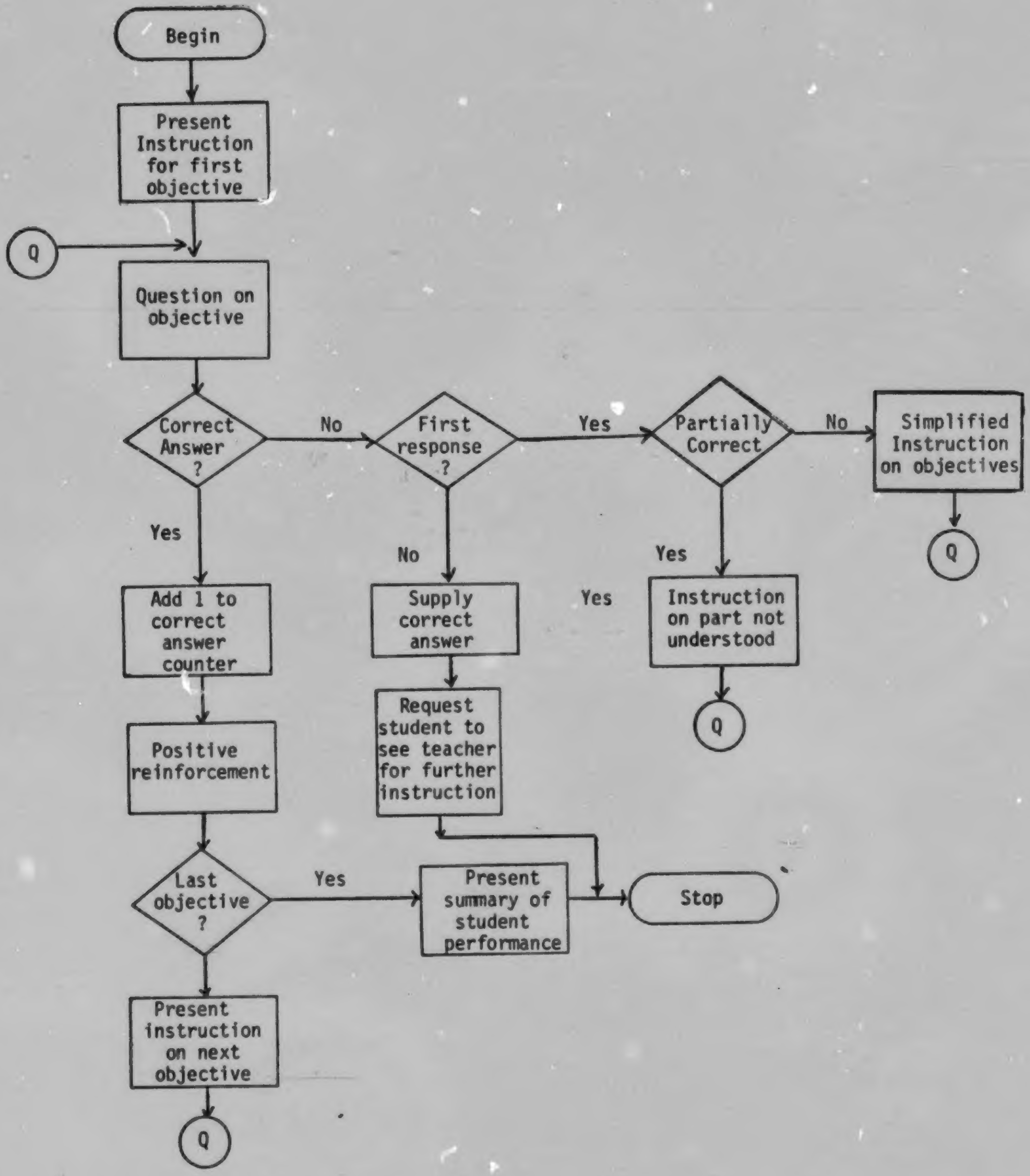

Begin
Present Instruction for first objective
Q
Question on objective
Correct Answer ?
No
First response ?
Yes
Partially Correct
No
Simplified Instruction on objectives
Q
Yes
No
Yes
Yes
Add 1 to correct answer counter
Supply correct answer
Instruction on part not understood
Q
Positive reinforcement
Request student to see teacher for further instruction
Last objective ?
Yes
Present summary of student performance
Stop
Present instruction on next objective
Q

Figure 2

Instructional Strategy - Simple Branching Sequence

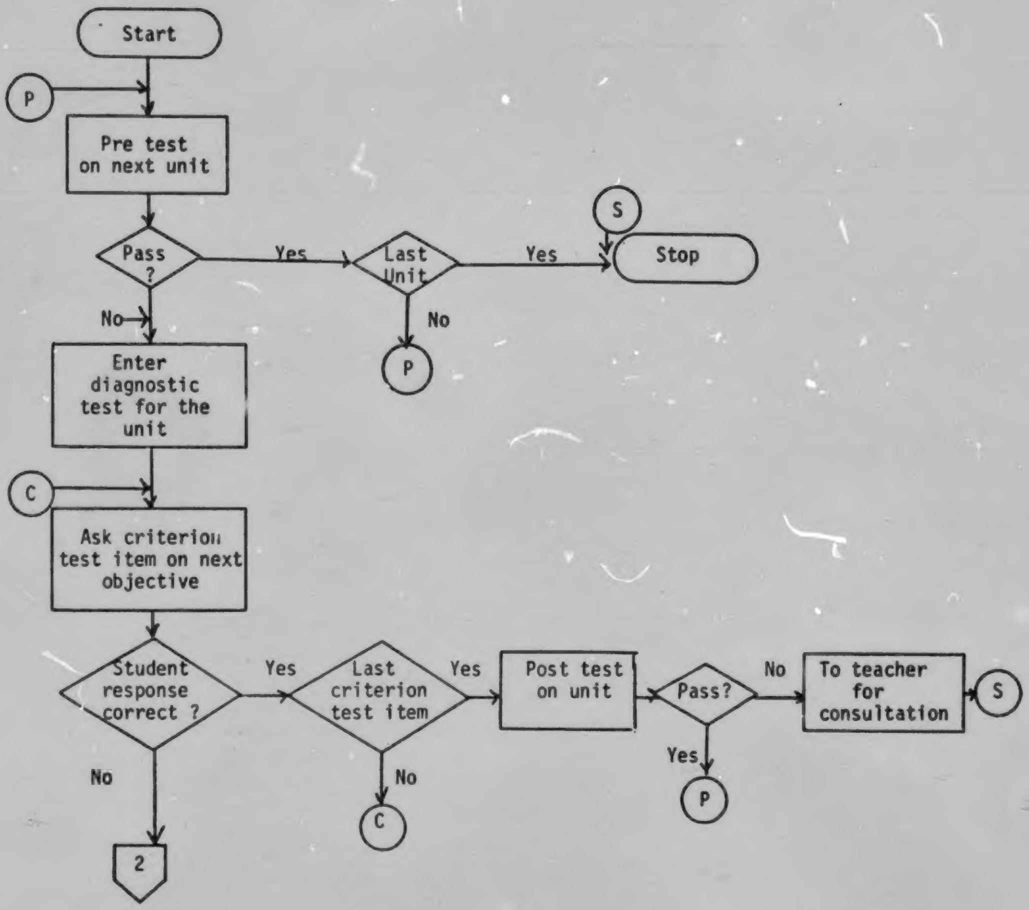

Start
P
Pre test on next unit
Pass ?
Yes
Last Unit
Yes
S
Stop
No
No
P
Enter diagnostic test for the unit
C
Ask criterion test item on next objective
Student response correct ?
Yes
Last criterion test item
Yes
Post test on unit
Pass?
No
To teacher for consultation
S
No
No
C
Yes
P
2

Figure 3

Instructional Strategy - Complex Branching Sequence

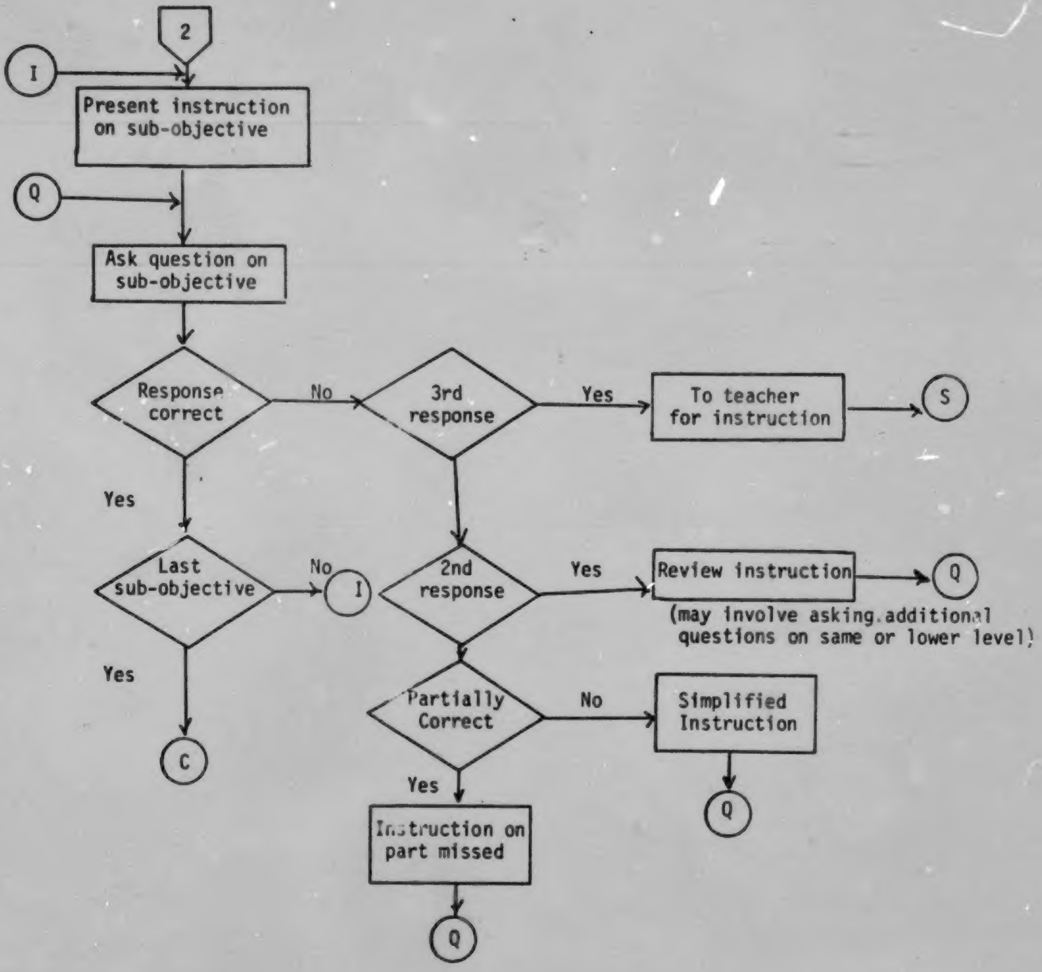

2
I
Present instruction on sub-objective
Q
Ask question on sub-objective
Response correct
No
3rd response
Yes
To teacher for instruction
S
Yes
Last sub-objective
No
I
2nd response
Yes
Review instruction
Q
(may involve asking additional questions on same or lower level)
Yes
C
Partially Correct
No
Simplified Instruction
Q
Yes
Instruction on part missed
Q

Figure 3 - Continued

Identify and Prepare AV Materials

Often additional media are required to communicate the instructional message to the student. The author should examine the lesson flow to determine which learning activities will be used to attain the objectives. Then the additional media to be used with those learning activities must be identified.

Additional materials could include:

1. Supplementary reading to avoid placing several screens of text for the student to read. Too often authors use the computer to simply present textual material to students. This practice is strongly discouraged.
2. Real objects like disk, tape, core rings, card decks, etc.
3. Audio tapes containing perhaps the teacher's explanation of a difficult concept.
4. A film, filmstrip, or videotape presentation of tutorial material.

However, keep in mind the following points when selecting additional media:

1. time required to develop audio-visual materials;
2. compatibility of media selection with the computer terminal in use;
3. appropriateness of the selected media to the activity--Is it better than any other media?

Due to the time involved in preparing additional materials, production of these materials should begin at this point.

Code Course Material

Now the author/programmer should be ready to write and code the course in an author language. There are numerous author languages available for coding instructional materials using the computer in an interactive mode. Among these are Coursewriter III, APL, Basic, Tutor, and Scholar. The language used at East Texas State University is IBM Coursewriter III. Coursewriter III, an IBM author language designed exclusively for CAI, is easy to learn and easy to use. Much of the processing logic is inherent in the operation codes (i.e. when the student answers a question correctly, control is automatically passed to the next problem unless routed elsewhere by the programmer.) Also, partial answer processing capabilities allow realistic interaction (eg. when two words are required as a correct answer, if one word is correct, the student can be told this fact and asked to change the other word.). Rather than expound upon the details of Coursewriter III, interested readers are advised to reference the IBM Coursewriter III Author's Guide.

Note that there is no perfect author language. Each has virtues and faults. If you have not selected an author language, some points to keep in mind are:

1. Is it just a computer language or is it designed for use in CAI?
2. Is it easy for both students and faculty to learn and use?
3. Is it transportable? (i.e. Can it be used on other computers?)
4. Is it supported by your computer vendor?
5. Have other educators used it?
6. What types of terminals are supported by the language?

No matter which author language is used to code the material, avoid the use of pat feedback to student responses; such as right, wrong, correct, incorrect. These kinds of pat responses to the student give CAI its "dehumanizing" reputation. Give hints, allow for unexpected answers, and provide a way to proceed to the next question other than by giving a correct answer. In other words, write a flexible program that will truly individualize instruction.

Nothing that has been written is perfect the first time. Much rewriting and recoding is necessary to develop a smoothly flowing program of instruction. Always keep the student in mind. Present the material in small, logical steps so that the student can establish closure with previously discussed points. The purpose and importance of each point toward the final goal should be stated. Use numerous break points for re-entry if the student signs off and review points if the student demonstrates confusion, or skip ahead if the student demonstrates complete mastery of a specific objective.

Debug

Unfortunately, very few computer programs execute successfully the first time. Therefore, time must be taken to debug the program; that is, get all of the programming errors out so that it will execute successfully. Unique to CAI programs is the need to search for misspelled words, sentence structure, and screen formatting errors, if cathode ray tube terminals are used. Often the answer processing statements do not function as originally anticipated. Thus, many trial executions must be completed to find all of the errors. Try all of the anticipated responses as well as a few unanticipated responses to test execution of all student paths through the program.

Peer Group Tryout

The evaluation phase is often overlooked by author/programmers. They tend to feel that once a program executes successfully with their test data, it is complete. For CAI programs, author testing is not sufficient. If the instructional package is to have substantial use by anyone other than you, peer group tryout is essential. Ask two or three fellow teachers in your department to take the course as students and make suggestions for improvement of the instruction or the answer processing. Teacher peers are valuable critics. Analyze their comments and if you feel that the suggestions are valid, revise the module. If additional coding is involved, then go back to step 6--coding the revisions.

Target Population Tryout

After the package has the "seal of approval" from teachers, it is time for the real test--students. Test the program with a small group of students having the characteristics of the target population. For instance, if the module teaches syntax of Fortran IV, test it with a section of Introductory Fortran IV students. Even though the program statements are syntactically correct, students very often do not perform as expected. Weak points in the instructional strategy or the programming will probably be identified. If the objectives of any module were not met, as indicated by student performance, major revisions may be necessary. If necessary, revise and recode.

For a true evaluation of the instruction, compare the competencies of those using CAI with the competencies of a group of similar students being taught the same material using other media. Such an evaluation procedure was conducted at East Texas State University by the author of this paper with respect to learning Coursewriter III. It was found that students who received tutorial CAI instruction learned the desired concepts as well as those involved in the traditional classroom environment, but in less time. An interesting fact was discovered with a third group who were exposed to both CAI and traditional instruction - this group out-performed the other two single-method groups by an average of 20%.

Review and Maintain

Finally, the package has passed all tests. Don't put it into production and then forget about it! Periodically review the material; analyze the comments made by the students concerning areas difficult to understand; and analyze students' unexpected wrong answers. You may need to continually revise various parts of the program to allow for individual student differences and to keep the content consistent with current advancements in the area of computer technology.

Documentation

One final point, documentation is extremely important not only to conserve time and effort on the material being developed so that other instructors may use it, but also as an aid to preparing additional modular units. When no documentation exists, the life of the package is doomed to be short. All of the following should be placed together for documentation:

1. definition of the topic (scope and sequenced tasks)
2. list of student characteristics and pre-skills (entry behaviors)
3. definition of specific objectives (in measureable terms)
4. flowchart of instructional strategy
5. audio-visual materials.

Conclusion

When one examines the large number of topics which are recommended for inclusion in computer science courses by curriculum committees, it is obvious that the computer science teacher needs all the help he or she can get. This is especially true with the new trends emerging in programming and design techniques. Thus, computer assisted instruction may well enable the computer science teacher to offer a more complete course if adequate application packages can be developed. And they can, if master teachers will take the time to follow each of the steps outlined. The net result of such effort will be more precise, yet comprehensive, instruction for the student.

THE COMPUTER -- A PERSONAL TOUCH

Dennis H. Sorge
Purdue University

ABSTRACT: For many years Purdue University has hand processed a report of students doing unsatisfactory work. In 1969 the Calculus committee of the Department of Mathematics wanted a more effective system for reporting to academic advisors calculus students doing unsatisfactory work on homework, quizzes, or tests or having excessive absence. The calculus reports in 1969 and 1970 still made too little use of a computer and too much use of clerical time. It was decided in 1971 that to be effective the system would have to be largely computerized. From Registrar supplied computer tapes the initial data base was created: a system for updating records was started; programmers began preparing software. With each new feature additional software and logistic problems had to be solved. Beginning Fall, 1973 all 12,000 enrollments in Computer Science, Mathematics, and Statistics were part of the data files. This program for monitoring students' progress influenced the development of a many faceted administrative computer system. Today the Division of Mathematical Sciences has a 95% computerized reporting system, an enrollment update system, a teaching assignment update system and other support operations. In Fall 1975 the University Provost's office requested that all Freshmen in all courses at the university be included in this delinquent student report system. Successful results on a larger scale would suggest a possible replacement for the university hand-prepared reporting system. About 5000 delinquencies were reported from a sample of over 40,000 enrollments. This close monitoring and subsequent personal contact is a reality only because of a computer.

It is not at all unusual that a college or university each semester should prepare a list of students who are doing unsatisfactory work. This is certainly the case at Purdue University. For many years this report of students doing unsatisfactory work, known as Delinquent Students Report, or yellow slips, has been a hand-processed report. Instructors were asked to pick up, complete and return the data gathering forms. To complete this mid-semester report of unsatisfactory work instructors were required to supply the student's name, social security number, university identification number and school of enrollment and sign the statement that the student's work was unsatisfactory. Many instructors felt that it was too much trouble. Many academic advisors felt that it supplied insufficient information and that the report arrived too late to be effective.

The decision to computerize

Partly because of these reactions the Calculus Committee of the Department of Mathematics decided in 1969 to develop a more effective system for reporting delinquent calculus students to their academic advisors. The committee did not know how the system was to be made operational but they did know that it should report information about the student's homework, quizzes, tests and absence. It was also decided the report should come out as soon as possible after the first examination which was about the fourth week of the semester.

To create the reports in 1969 and 1970 instructors were asked to supply the names of the students who had turned in less than half of their homework, who had scored less than half on quizzes, and who had excessive absence. The office responsible for producing the report, known as the Calculus Office, already had test results since it regularly maintained records on students' scores. Three clerical people spent almost three weeks collecting and sorting data and preparing reports that listed students alphabetically within their school of enrollment. Unfortunately, this procedure only solved one of the three problems associated with yellow slips. More information was reported but it was still cumbersome for the instructors to supply and it still took too long to generate.

To simplify the reporting procedure it was decided that instructors would be sent preprinted lists containing the names of their students, the social security numbers and the school of enrollment. However, to do this required that the Calculus Office keypunch all the classlists for the calculus courses and, using a computer program, sort and print these lists. This succeeded in simplifying the reporting process somewhat for instructors but it created an almost impossible burden on the Calculus Office since there are usually 4,000-5,000 calculus students at Purdue.

Investigations into how to eliminate the need to keypunch classlists led to the discovery that the Registrar's Office would supply computer tapes containing the information needed. With this support it was decided in 1971 that to be effective this system must be largely computerized.

Building a data base

Initial contacts were made with the Registrar's Office to ascertain what types of data tapes could be obtained from them. Tape formats and loading problems were discussed. The first stage of developing this reporting system was to write programs which would create classlists on which instructors could record delinquencies and so that the Calculus Office would not have to keypunch this information as they had done previously.

By the fall of 1972 we succeeded in overcoming the problems resulting from computer incompatibilities. Tapes were created on IBM 370 equipment; the Calculus Office was building its files using Control Data 6600 hardware; tapes in 6000 character blocks had to be split to avoid special tape drives; 800 bpi density was reduced to 556 bpi to eliminate superious parity errors. Computer software extracted names and social security numbers from one tape and course and division of enrollment from another. It then produced a punched card containing this information. Another program produced lists of student names and column headings under which instructors could mark if students had turned in less than half their homework, scored less than half on quizzes or whether they had been absent excessively from class. When this information was returned to the Calculus Office the cards for those students reported as delinquent were removed from the initial decks and sorted by school of enrollment and by course.

These cards then were used to generate a report to counseling offices. This report had a lead paragraph, the students' names and social security numbers listed by a program. But it was still necessary to mark by hand areas in which the students were reported as being delinquent. It was now easier for instructors to report delinquent students. However, the amount of paperwork for the Calculus Office was still too great.

The difficulties with this system had to be improved. There must be some way for instructors to report student delinquencies that could be mechanically processed. It should also be possible to produce lists for academic advisors without requiring any hand work.

At this stage a critical assessment was necessary and major decisions had to be made. They were as follows: The entire system must be reassessed and in fact completely rewritten and redeveloped. The small patchwork FORTRAN programs would be eliminated. COBOL programming would be used because its self documentary nature provides for easier transition from one programmer to the next and because its file definition and manipulation procedures were appropriate for the system being developed. A 95% computerized operation would be the goal. A study of input media and available hardware led to the decision that data would be collected from instructors by using a semi-mark sense computer card. The university had facilities for sensing and punching the marks into the card. Procedures and equipment that would permit the use of optical scan sheets were not available. Computer programs would create initial data banks and from these data banks the semi-mark sense cards would be punched with the student's name and class identification. Appropriate places would be darkened for any delinquencies to be reported. The cards were ordered and the programming began.

As the system was developed and the programming progressed it became obvious that there were additional problems that had to be solved. One of the items reported on delinquent reports is the student's test scores. For some time the Calculus Office had been collecting information about test results. If this information were in a form suitable for computer input, it could be fed into the data banks so that it would no longer be necessary to ask instructors about test results. Since there are always exceptions to this, the option of reporting test scores using the semi-mark sense card must be maintained. Also necessary was the development of a system for grade changes. It was important that if a student dropped a course, he would not be reported as delinquent. This brought up the entire subject of a continuous updating of the data files. All dropping and adding of courses would have to be recorded and all changes within courses would have to be determined.

The system grows

In the spring of 1973 the first nearly totally computer generated report was produced. However, many of the problems already discussed had not yet been solved. Record updating was still not functioning. We had dealt with hardware incompatibilities, with software problems and now we had to handle administrative difficulties.

The system would not be efficient unless all student enrollment changes could be recorded and processed in one location. Since distribution of mark sense cards required information about instructors the entire area of computerized record keeping on teaching assignments began to develop.

The spring 1973 report generated for the calculus courses was so well received by academic advisors and instructors that there were requests to include in the reporting system courses other than the calculus ones. Because of administrative responsibilities of personnel involved in developing the system and because of the nature of the growth of the system it was decided that the data banks would contain all students enrolled in any course in Mathematics, Computer Science or Statistics at Purdue University. This meant that up-to-date data tapes on 15,000 enrollments would be maintained. During the fall semester 1973 two reports were produced in the first eight weeks of the semester. Approximately 6,000 students were being monitored for this report. Several hundred of these were reported as delinquencies. Getting this far, however, had required nearly four years devoted to the development of software and logistic patterns conducive to effective operation. This alone would perhaps not have made the program successful had it not been for adequate administrative support. This support encouraged instructors to maintain accurate and up-to-date records of their students' progress. With accurate records it was now possible to reduce instructor reporting time to not more than ten seconds per delinquent student. The Calculus Office had reduced its non-computer work to the distribution of cards, the collection of cards, and the mailing of the final reports.

The administrative system that had developed out of this request for reporting delinquent students was yet not fully developed. There were still logistic rough spots. The problem of record updating had to become a smooth, routine operation. Enrollment changes alone number 3,000. Procedures were established and printed in a checklist format.

With these problems solved, the 1974-75 academic year proved to be a very successful reporting year. Instructors were routinely asked twice during the semester for information about their students. The number of students involved had now grown to approximately 7,000. Whereas in 1969 it had taken three clerical people nearly three weeks to complete this report, the clerical time now had been reduced to little more than two hours for one person.

The reports were praised by academic advisors across the campus. They found them very helpful because they contained sufficient information and they came at a time when changes in enrollment or study habits could still be brought about. Likewise instructors had no complaints because reporting delinquent students was now a very simple procedure. The depersonalizing effect often attributed to the computer had made it possible for instructors and advisors to show and successfully communicate a personal concern for their students' academic success.

Research during the 1973-74 academic year showed that the report was successful in getting many students to improve the quality of their work or in getting many students to drop courses before receiving a failing grade. The research also showed that every student who received a failing grade in one of the courses for which delinquent reports were produced was notified at least once during the semester that his work was unsatisfactory.

There were several significant side effects resulting from the development of this reporting system. Once there were as many as 700 students out of 15,000 who were not attending the division of a course in which they were enrolled and never located in their specific class. With this system and record updating it was possible within a year's time to reduce that number to approximately 25. Additionally, a data retrieval system was developed which took the initial tapes supplied by the Registrar and produced many, many forms of assorted output. Research that was once not possible because of the inability to obtain the data in a desired form now became routine.

University applications

General university-wide dissatisfaction with the 'yellow-slip' reporting system mentioned earlier caused the university Provost's office to request that the Calculus Office, now called the Office of Undergraduate Services, include in its reporting system on a trial basis all students who are freshmen at the university.

During the fall semester 1975 nearly 40,000 semi-mark sense computer cards were produced and distributed to faculty across the university. All freshmen students in all courses were now subjects for delinquency reports. Nearly 5,000 delinquencies were reported via these semi-mark sense cards and at about the seventh week of the fall semester a report was generated and sent to academic advisors. The problems that presented themselves during this extension were distribution, collection and definition of terms -- homework, quizzes, test. None of the problems were computer-related.

The total system was not used in this reporting, only that part of it related to the reporting of delinquent students. So, in some cases, students who had dropped courses may have been reported as being delinquent. Nevertheless, this did not detract from the effectiveness of this program. Many additional difficulties surfaced at this time. Among these were that many instructors did not keep accurate and up-to-date records; many students were not evaluated until nearly half-way through a semester; and even with this simplified reporting system some instructors did not wish to bother. Nevertheless, the success of the program was far greater than its faults.

It was apparent that the logistics involved in handling nearly 40,000 computer cards is a greater problem than it should be. But since the university Registrar's office decided that they would begin experimenting with this system and would use their new facilities for optical scanning to gather data, this problem would be solved. Semester grade reporting, already an optical scan procedure at Purdue, would serve as an excellent media for the delinquent student report. Format changes seemed to be the only necessary variations.

Almost all of the computing problems in this system have been solved. The major concerns now involve administration and faculty. When additional cooperation and improvement in record keeping and evaluation is brought about the reporting system will show its greatest potential. Successful experimentation with selected classes and optical scan reporting sheets will result in university-wide replacement of the ancient yellow slip hand-processed reporting system.

One factor that must always be concerned in the development of any system is the cost. The computerized system of reporting delinquent students used for monitoring and subsequently reporting all delinquent freshmen students during the fall semester 1975 costs only about $300. This cost included cards, computer time and student help for distribution and collection. Operational costs university-wide should be in the $2,000-$3,000 range. Developmental costs spent from 1969 to the present were about $5,000.

This is a very small cost when compared to the expense for students and the university resulting from students unsuccessfully completing a course. Most important of all, by using a machine to store and rapidly analyze large quantities of information, to temporarily remove the human element, has permitted instructors, advisors and administrators to show more concern, guidance and consideration for students. In this area at a large university the computer has contributed to a personal touch.

THE FUTURE OF COMPUTER ASSISTED INSTRUCTION: A SUMMARY AND ANALYSIS OF THE CARNEGIE COMMISSION REPORT - COMPUTERS AND THE LEARNING PROCESS IN HIGHER EDUCATION

C. Edward Streeter
Professor of Information Sciences
Illinois State University

ABSTRACT: The purpose of this paper is to stimulate thinking and discussion about the misuse and potential of computer assisted instruction. The central focus of the paper is an examination of existing and potential computer support to instruction based on an analysis of the learning process. A look is taken at the effectiveness of these uses in terms of possible equally effective but less costly alternatives.

26

This paper is a summary and analysis of a report prepared for the Carnegie Commission on Higher Education, Computers and the Learning Process in Higher Education by John Rockart and Scott Morton of the Sloan School of Management, Massachusetts Institute of Technology.

Before going into the report itself, let me give you some background information that will help you understand the biases from which this summary and analysis is done. I spent my entire public school career 1948-1964 in Iowa as a math teacher, coach, principal, superintendent, curriculum and instructor consultant. I was working as curriculum-instructor consultant in Keokuk, Iowa, when offered an NDEA Instructional Technology Fellowship to work on a doctorate at Michigan State University. My reaction was, now I will learn what I am supposed to know in my work as an instructor consultant. After three years of study and eight years of trying to apply what I had learned, I have come to one basic conclusion: There is a lot we do not know about how people learn and how to instruct.

My interests lie in the utilization of all types of instructional media, and particularly in the development of models and schemes for selection and utilization of instructional media. My present teaching/research interest is the appropriate use of various instructional media in teaching a computer literacy course in two formats: regular classroom and guided independent study. As we get into the work of Rockard and Morton, you will see my interest in preparing this paper and looking forward to interacting with you about it.

Rockart and Morton focus on computer supported instruction (Computer Assisted Instruction (CAI) in the broad sense) as opposed to computer managed instruction or instruction about computers. Their real concern, however, appears to be the development of a model for the appropriate selection of all types of instructional media; a concept that is needed for all levels of teaching, not just higher education.

Their report comprises ten chapters and 356 pages, including an appendix. The first part of the book is introductory and background material. Three important questions relating to effective CAI utilization are presented, followed by a discussion of the teaching and learning theory, CAI hardware and courseware, and some descriptions of CAI projects. Except for the three questions, this background material will not be covered in this summary.

The three questions they suggest need answering are: (1) Can CAI be cost effective?, (2) In what specific ways will the computer effect the learning process in higher education? and (3) How should a faculty member approach the selection of one or more of these types of CAI for a particular course of study?

Chapter six is the "heart" of the report, development of a plan to match technology to learning. While Chapter six reports what the authors thinkought to be, the next chapter reports on what is, using a 1974 survey of CAI in Massachusetts higher education. The latter part of the book examines the environmental and internal constraints affecting CAI adoption in higher education, the author's conclusions, and the appendix.

My summary and analysis deals mainly with the development and application of Rockart and Morton's system for selecting the most appropriate uses of computers in the learning process. They start with the general learning model as shown in Figure 1.

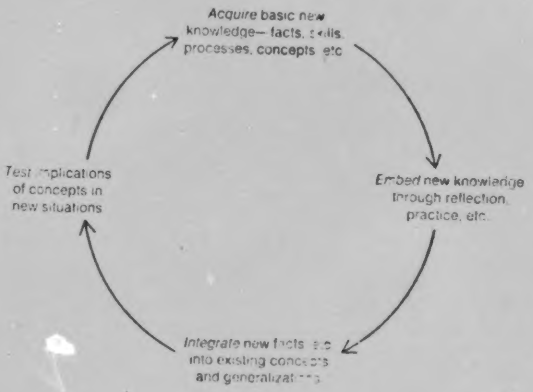

Acquire basic new knowledge—facts, skills, processes, concepts etc
Embed new knowledge through reflection, practice, etc.
Integrate new facts etc into existing concepts and generalizations
Test implications of concepts in new situations

Figure 1

In the initial stage, acquisition, the learner is exposed to basic data, skills, and concepts. The second stage, inbedding, has the student practice and/or ponder his newly acquired knowledge or skill. During the integrating stage, he assimilates the new information into his existing cognitive structure of the world, and the final stage is testing his learning in new situations: Concrete or simulated.

The other element of the learning matrix, shown in the left column of Figure 2, is a categorization of materials (Course content). These four categories are described as follows:

1) Facts, including definitions and other basic information relating to specific single items or ideas.

2) Skills, including both procedures and rules and their applications.

3) Established concepts, as well as other theories, hypotheses, postulates or assertions that are well enough established to be of no interest to the current researcher in the field. "Concepts" use "facts" as basic building blocks.

4) Frontier concepts, including not only recent development but also long standing issues which either have not been or cannot be resolved.

	Acquisition	Embedding	Integration & generalization	Testing in new situations
Facts and definitions				
Skills and procedures				
Established concepts				
Frontier concepts				

Figure 2

The next major step in the development of their scheme is a grouping of available learning mechanisms described in an earlier chapter; these are as follows:

Audio:	lecture, case study, class discussion, tape, radio.
Visuals:	visual aids, film, video tape, cable TV.
Written:	textbooks, programmed instruction, study modules, written assignments.
Computer:	tutorial, drill and practice, problem solving, inquiry, simulation, games.
Other:	laboratory experience, real world experience.

Sixteen attributes are then identified which each of the above learning mechanisms contains to some degree. These attributes are categorized and set forth in Figure 3. Both the categories and the attributes are somewhat self-explanatory, and time will not permit a detailed discussion here. A cross-section of MIT faculty were asked to rank each of the learning mechanisms in terms of the degree each mechanism possesses the various attributes.

A. *Content related*
   1. Ability to telescope time
   2. Ability to present structure
   3. Provision of a rich environment
   4. Ability to provide ill-structured material
   5. Flexibility for adding new material quickly
   6. Support for the learners' structured, clerical tasks
   7. Support for unstructured data manipulation

B. *User related*
   8. Degree of learner control
   9. Ability to adjust to individual learner needs
   10. Ease of use

C. *Communications related*
   11. Amount of sensory impact
   12. Amount of emotional impact
   13. Degree of learner feedback
   14. Ability to access data or concepts previously learned

D. *Economics*
   15. Low cost per data item or concept
   16. Decentralized availability

Figure 3

Figure 4 shows the average of the professors' ratings: one (1) being most favorable, and ten (10) being least favorable. The results most germane to this audience are those listed under the heading computer, and this discussion is limited to that group. To summarize and analyze the data, we shall consider a score of three or less as being positive for effective utilization, a score of eight or more as negative, and four through seven as neutral. The results of this analytical format are set forth below with neutral items not listed. Under tutorial positive there are only two items: the degree of learner feedback and decentralized availability. Under negative, however, there were eight: ability to telescope time, provision of rich environment, ability to provide ill-structured material quickly, flexibility for adding new material quickly, support for the learner's structured and clerical tasks, support for unstructured data manipulation, amount of sensory impact, and amount of emotional impact.

Drill and practice found five positive items. They were, ability to telescope time, support for the learner's structured and clerical tasks, degree of learner control, degree of learner feedback, and decentralized availability. Under drill and practice negative there were also five items. These were, provisions for rich environment, flexibility for adding new material quickly, ability to provide ill-structured material, amount of sensory impact, and amount of emotional impact.

As we moved into problem solving, there were eight positive items: ability to telescope time, support for the learner's structured and clerical tasks, support for unstructured data manipulation, degree of learner control, ability to adjust to individual needs, degree of learner feedback, ability to access data or concepts previously learned, decentralized availability; while problem solving found only five negative features: flexibility for adding new materials quickly, ability to provide ill-structured material and clerical tasks, amount of sensory impact, emotional impact, low cost for data items or content.

There are six positive items under inquiry. There are provisions for rich environment, ability to provide ill-structured material, degree of learner control, ability to adjust to individual learner needs, ability to access data or concepts previously learned, decentralized availability. Under negative there are six items--ability to telescope time, ability to present structured materials, flexibility for adding new materials quickly, support for unstructured data manipulation, amount of emotional impact, low cost data items or concept.

As we move into the more sophisticated uses of computers such as simulation games, we find many more positive aspects as compared to the negative. Under simulation, we have eleven positive aspects. They are ability to telescope time, provisions for a rich environment, ability to provide ill-structured material, support for learner structured and clerical tasks, support for unstructured data manipulation, degree of

	Attribute															
	Content							User			Communication				Economics	
Mechanism	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Audio																
Lecture	5	3	8	1	1	10	10	8	1	5	5	3	8	10	3	8
Case study	5	10	3	3	5	10	10	8	5	5	5	5	8	10	5	10
Class discussion	5	10	3	3	1	10	10	8	8	5	5	5	3	10	5	10
Tape	5	5	8	5	3	10	10	5	10	5	5	5	10	5	5	3
Radio	5	5	8	5	3	10	10	10	10	3	5	5	10	10	5	3
Visual																
Visual aids	5	3	10	10	1	8	10	10	8	5	3	5	10	8	5	10
Film	1	3	5	1	10	8	10	8	10	8	1	1	10	8	8	5
Videotape	1	3	5	1	5	8	10	8	8	8	1	1	10	8	8	5
Cable TV	1	3	5	1	5	8	10	10	10	5	1	1	10	10	8	3
Written																
Textbooks	5	3	8	3	10	10	10	10	10	1	8	8	10	5	1	1
PI	10	5	10	10	8	10	10	10	8	1	8	10	3	8	3	1
Study modules	10	3	10	5	8	8	10	8	8	3	8	8	5	8	3	1
Written assignments	10	5	10	10	5	10	3	8	8	1	8	8	3	8	10	1
Computer																
Tutorial	10	5	8	10	8	10	10	5	5	5	8	10	3	5	5	3
Drill and practice	3	5	8	8	8	3	5	3	5	5	8	8	1	5	5	3
Problem solving	1	5	5	8	8	3	1	3	3	5	8	8	1	3	8	3
Inquiry	8	8	1	1	8	5	8	1	3	5	5	8	5	1	10	3
Simulation	1	5	1	1	8	1	1	1	3	5	3	5	3	3	10	3
Games	1	5	3	1	8	1	3	3	3	5	3	3	5	5	10	3
Others																
Laboratory experiments	5	5	1	10	8	10	10	3	10	8	5	8	1	10	10	10
Real-world experience	10	10	1	10	5	10	10	3	5	10	3	3	5	10	10	8

Figure 4

learner control, ability to adjust to individual learner needs, amount of sensory impact, ability to access data or concepts previously learned, and decentralized availability. The negative features of simulation are limited to two: flexibility for adding new material quickly and low cost for data item or concept.

Under simulation, amount of emotional impact was neutral, while under games it is classified as a positive aspect. The reverse is true for learner feedback and ability to access concepts previously learned. They are both three's in simulation, and under games they are rated five.

Rockart and Morton used the attribute data to map the mechanisms' attributes on the learning matrix. To carry out this mapping, the attributes were assigned into new categories with some attributes put in more than one of the new groupings. Figure 5 shows the author's opinion of how these new categories cover certain areas of the learning matrix.

In Area I and II, the C-E and FDBK represent cost effectiveness and feedback. In Area III, L/CTL represents learner control and assistance, and MANIP represents data control and manipulation. In Area IV, EMUL stands for real world manipulation, and in Area V, ADAPT stands for adaptation.

The next step was to look at the attribute ratings of each learning mechanism and draw some conclusions.

The investigators' analysis of the data from the mapping is as follows:

1. Emphasizing one mechanism is a mistake; we should maintain a smorgasbord of learning mechanisms.

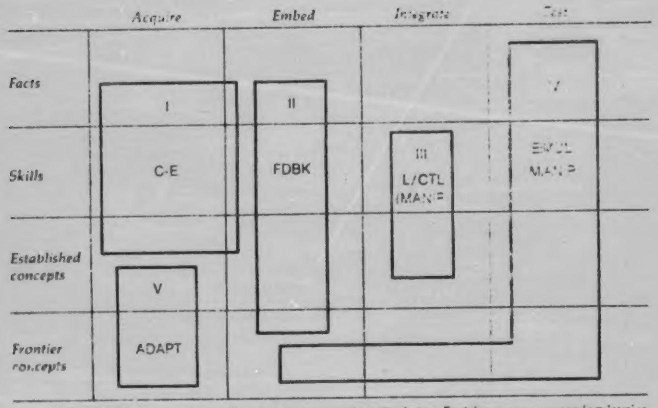
2. On figure 5, areas I and V (cost effectiveness and adaptability), traditional learning mechanisms (books, program texts, lectures, etc.) appear best suited and should not be computerized except for remedial learning.

3. In area II (feedback), because of decreasing computer costs we can expect computerized practice to make major inroads on traditional pencil and paper methods.

4. In areas III and IV (learner control and assistance, real world emulation and data recall and manipulation), most existing learning mechanisms are inadequate and we can expect to see tremendous growth in computerized simulation, games, inquiry and problem solving.

After defining what they thought professors should be doing with computers, Rockart and Morton surveyed Massachusetts higher education (1974) to see what was being done. This survey elicited the conclusion that in general computer power for teaching with and about computers has and will continue to grow. In addition, they found that the growth pattern was quite similar to their normative model with most efforts in areas II and III, and the use of simulation, gaming, and inquiry less common.

After examining the environmental conditions affecting higher education (attitude of society, economic conditions, funding agencies, student population, competition for students, and student



NOTE: Feedback is an essential ingredient in learner control manipulation. Enrichment is at a secondary level in almost all areas.

Figure 5

attitudes), they concluded society will (but not to the degree of the past decade) continue to support higher education, student populations will grow, competitive outside forces in economics will move us toward more uses of technology, and a more sophisticated student body will be more receptive to the use of instructional technology.

The analysis of the internal constraints was not as positive. While the administrators attitude may be favorable, the faculties attitudes will not have the same positive impact. The average faculty member will see many more reasons not to work towards the introduction of instructional technology than positive factors.

The overall conclusions of the Rockart, Morton report are, first, the change and improvement of instructional technology will have a potential impact on the way we conduct outselves in high education.

> The three major influences appear to be computers (reduction of cost, minis and networks), television (voice interaction with remote audience) and libraries in their nonpaper form.

Second, before putting more money and effort into the utilization of technology, we need to better understand where and why to apply it. To underscore this statement, they set forth the following specifics:

> We need a robust model of the learning process.
>
> We need a "smorgasbord" of learning mechanisms.
>
> Programmed instructional type CAI has had too much emphasis.
>
> The opportunities for computers in high education lie in the "enrichment activities."
>
> In most cases we will be adding new learning mechanisms, not replacing traditional ones.
>
> One case where replacement can and should take place is computerized drill and practice replacing traditional homework and /or written assignments.

And, finally, the pace of implementation of CAI will be slow because

> the favorable forces are not nearly as powerful as the negative.

In doing my analysis of this report, it seems appropriate to start with the three questions posed by the investigators and the answers they provided. At the time the questions were set forth, they indicated that only to the first one--Can CAI be cost effective?--was there an unambiguous answer. They felt CAI could be cost effective in some cases, and later conclude that drill and practice and simulation are cost effective but tutorial is not except in unique situations such as remedial learning.

In trying to answer the other two questions--In what ways will computers affect higher education, and, How should a faculty member approach selection of CAI? they certainly made a major contribution toward an answer with their scheme for placement of learning mechanisms in various stages of the learning process. But as we all know, there are no simple solutions to complex questions. They did take a major step toward providing us with tools and background to develop plans for selection and utilization of computer assisted instruction but, at least in my judgment, they may not be totally correct. Rather than deal with whether or not they provided definitive answers to these rather global questions, I prefer to analyze their specific overall conclusions.

Let me start with those specific points that I agree with almost in total: technology will continue to have an impact on instruction in higher education, the adaptation pace will be slow because of resistance at the faculty level, we need some theoretical models of learning upon which to base our adaptation, we will never have a single or even just a few learning mechanisms.

I agree about 75% that the best opportunities lie in the enrichment categories, that CAI drill and practice can conceivably replace traditional written homework assignments, and the programmed instruction approach to CAI has seen its day.

My basic disagreements lie in two areas, and both require some crystal ball gazing. One, I do not see large computer networks such as proposed by the PLATO and others as being likely to have long range success but necessary intermediate steps. It appears to me that the developments of intelligent terminals, micro processors, and video disc storage are going to provide us with very sophisticated self-contained individual teaching stations that will avoid the reliability and storage problems of a centralized network.

My second and more serious disagreement is with the learning model and matrix proposed. Although I concede it is better and more operational than anything else I have come across. The problem may be in what we are trying to accomplish in higher education. Are we imposing rural school educational goals and procedures on students living in a highly sophisticated communication and information processing society? So far, our utilization of TV and computers in instruction have been to put old wine in new bottles. I do not know the answer, but maybe we need an educational "think tank." Just as the military has changed its goals and procedures in fighting a war, maybe some of the things we have held sacred in education can be bypassed, or at least learned in a much different way.

For the interim in which you and I will have to operate on a day-to-day basis, I plan to operate from a model and matrix similar to what Rockart and Morton propose, and try to use a problem solving approach to the selection of learning mechanisms rather than become a proponent of a single technology.

BIBLIOGRAPHY

Rockart, John Fralick and Michael S. Scott Morton. The Carnegie Commission on Higher Education Report: Computers and the Learning Process in Higher Education. New York: McGraw-Hill, 1975.

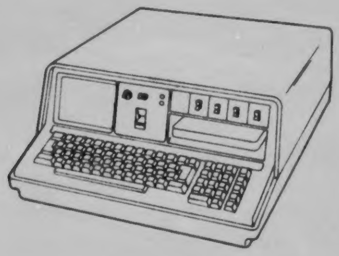
USING APL FOR DRILL & PRACTICE AND CAI

Richard F. Conklin

IBM Corporation
255 Alhambra Circle
Coral Gables, FL 33134

ABSTRACT: The APL language is well-known for its problem-solving power in many applications, particularly education. It offers the student a "math laboratory", in which to experimentally define and check out various computer functions. Additionally, it provides the teacher with a tool for tailoring special drill & practice and computer-aided instruction routines to reinforce and test material covered in class. This paper describes a few of these techniques.

At the time this paper was submitted for consideration, I had developed several APL "CAI" functions on an IBM System/370 time-sharing system (APL/CMS running under VM/370). Shortly afterwards, I joined IBM's 5100 Portable Computer marketing force and was pleased to discover that all of the routines could be easily "dumped" on the 5100 and run with no modification. (There was one . . . I replaced the word "percent" with "%", a special character not previously part of the APL keyboard.) I feel that the combination of APL and a full-function desk top computer yield a "teaching machine" with endless possibilities.



IBM's Portable Computer

A few simple APL techniques are essential to writing CAI-type functions. In this example, the built-in random number generator is used to produce two numbers, X and Y, which the student is asked to add.

```
      ∇ CAI1 RANGE;X,Y
[1]    START:X←?RANGE
[2]     Y←?RANGE
[3]    GUESS:X,'+',Y,'=?'
[4]     →(⎕≠X+Y)/WRONG
[5]     'RIGHT!'
[6]     →START
[7]    WRONG:'TRY AGAIN..........'
[8]     →GUESS
      ∇
```

On line number 4 the student response is compared with the true sum of X and Y and if not equal (≠) a branch is executed to the line labelled WRONG. This is the general form for a conditional branch in APL:

→(condition) / LABEL

Trying the function, we discover that the only way out is a branch out (→) in lieu of an answer.

```
      CAI1 20
6+10=?
⎕:
      16
RIGHT!
10+20=?
⎕:
      15
TRY AGAIN..........
10+20=?
⎕:
      30
RIGHT!
3+4=?
⎕:
      →
```

Recent enhancements in the language offer more possibilities for CAI. Student responses can be accepted as character input, and analyzed for typos, special commands or formats before converting the string to numeric for grading.

The following function allows any APL expression to be used as a model for a drill & practice session. Each constant in the model equation becomes an upper limit for a randomly-generated number. The word END is used to break the loop, and a score is printed. Some examples follow, and the listing of the function may be found at the end of this paper.

```
      CAI2 '10×10'
TO STOP, TYPE:    END
4×7
28
VERY GOOD
2×7
15
NO, TRY AGAIN
2×7
END
SCORE: 50%
      CAI2 '10+10+10'
TO STOP, TYPE:    END
7+9+3
19
VERY GOOD
5+4+3
12
VERY GOOD
1+10+5
END
SCORE: 100%
      CAI2 '100⌈100'
TO STOP, TYPE:    END
15⌈95
95
VERY GOOD
42⌈14
14
NO, TRY AGAIN
42⌈14
END
SCORE: 50%
```

The format for performing a conditional branch in APL can be a bit cumbersome when used repeatedly in a CAI or drill & practice routine. A better way is to write a small "one-liner" function which compares a student response with a canned answer. Called CHECK, it can be used whenever student responses need to be evaluated. Shown below, it is used within function CAI3:

```
    ∇ BRANCH←CHECK ANSWER
[1]   BRANCH←∧/ANSWER=(⍴ANSWER)↑⍞
    ∇
      ∇CAI3[⎕]∇
    ∇ CAI3
[1]   'WHO WAS BURIED IN GRANT''S TOMB?'
[2]   →(CHECK 'GRANT')/RIGHT
[3]   'SORRY, WRONG ANSWER'
[4]   →1
[5]  RIGHT:'VERY GOOD!'
    ∇
      CAI3
WHO WAS BURIED IN GRANT'S TOMB?
LINCOLN
SORRY, WRONG ANSWER
WHO WAS BURIED IN GRANT'S TOMB?
GRANT
VERY GOOD!
```

Still another technique for creating CAI drills involves "packaged" APL functions which require no programming knowledge of the teacher. One, called ENTER, asks for each of four key elements: the question, the correct answer, a correct answer message, and a hint (wrong answer message).

```
      AEDS←ENTER NEW
Q 1:   WHAT STATE IS PHOENIX IN?
A:     ARIZONA
RESP:  VERY GOOD!
HINT:  IT BEGINS WITH 'A'

Q 2:   WHAT HOTEL ARE WE IN?
A:     ADAMS
RESP:  EXCELLENT.
HINT:  WHO WAS OUR 2ND PRESIDENT?
```

The APL variable, AEDS, is used to store the drill. Any number of questions and answers can be entered, and a null (blank) line terminates the function.

Before the first student takes the drill, the teacher may choose to reset the scorekeeping function. SETUP AEDS does this.

```
      SETUP AEDS
      DRILL AEDS
WHAT IS YOUR NAME?  SUSIE SMART
Q 1:   WHAT STATE IS PHOENIX IN?  ARIZONA
VERY GOOD!
Q 2:   WHAT HOTEL ARE WE IN?  ADAMS HOTEL
EXCELLENT.
THANK YOU, SUSIE   THAT'S ALL FOR NOW.
      DRILL AEDS
WHAT IS YOUR NAME?  DENNY DUMMY
Q 1:   WHAT STATE IS PHOENIX IN?  IDAHO?
IT BEGINS WITH 'A'  ALABAMA
THE CORRECT ANSWER IS: ARIZONA
Q 2:   WHAT HOTEL ARE WE IN?
COME ON, TRY SOMETHING!  HOLIDAY INN
WHO WAS OUR 2ND PRESIDENT?  JOHN ADAMS
EXCELLENT.
THANK YOU, DENNY   THAT'S ALL FOR NOW.
```

Finally, when the last student has taken the drill, the teacher displays the class statistics, using function SCORES:

```
      SCORES
2 QUESTIONS, 2 STUDENTS.
CLASS AVERAGE: 62.5%

STUDENT:          SUSIE SMART
QUESTION NO.      1 2
POINTS (2=MAX):2 2
AVERAGE:          100%

STUDENT:          DENNY DUMMY
QUESTION NO.      1 2
POINTS (2=MAX):0 1
AVERAGE:          25%
```

```
      ∇CAI2[⎕]∇
    ∇ CAI2 FORMULA;NUMS;BIF;P;DRILL;ANS;COUNT
[1]    COUNT←''
[2]    'TO STOP, TYPE:   END'
[3]    NUMS←FORMULA∊'0123456789'
[4]    BIF←FORMULA[P←(~NUMS)/⍳⍴FORMULA]
[5]    FORMULA[P]←' '
[6]   LOOP:DRILL←⍕?⍎FORMULA
[7]    DRILL[(DRILL=' ')/⍳⍴DRILL]←BIF
[8]   AGAIN:DRILL
[9]    →(∧/'END'=3↑ANS←⍞)/DONE
[10]   →((⍎DRILL)=⍎ANS)/OK
[11]   'NO, TRY AGAIN'
[12]   COUNT←COUNT,0
[13]   →AGAIN
[14]  OK:'VERY GOOD'
[15]   COUNT←COUNT,1
[16]   →LOOP
[17]  DONE:'SCORE: ',100×+/COUNT÷⍴COUNT,⎕AV[173]
    ∇

      ∇ENTER[⎕]∇
    ∇ Z←ENTER OLD;Q;HRA;F
[1]   NEXT:⍞←F←'Q',(2 0⍕1+0.25×+/OLD='⍺'),': '
[2]    →(0=⍴Q←7↓⍞)/END
[3]    HRA←(0⍴⍞←'RESP:   '),'⍺',7↓⍞,'⍺',0⍴⍞←'A:
[4]    HRA←7↓⍞,(0⍴⍞←'HINT:   '),'⍺',7↓⍞,HRA
[5]    OLD←((¯1×OLD[⍴OLD]≠'⍺')↓OLD),F,Q,'⍺',HRA
[6]    ''
[7]    →NEXT
[8]   END:Z←OLD,'⍺'
    ∇
      ∇DRILL[⎕]∇
    ∇ DRILL X;F;N;C;R;H;T;NA
[1]    NAME←NAME,NA←20↑⍞,' ',0⍴⍞←'WHAT IS YOUR NAME?
[2]    NR←NR,(T←0)+⍴NA
[3]   S:→((1↑X←T↓X)='⍺')/END
[4]    ⍞←F←(¯1↓X[⍳T←X⍳'⍺'],0⍴SC←SC,2),'  '
[5]   A:→(0=⍴N←(⍴F)↓⍞)/TRY
[6]    H←¯1↓X[⍳T←(X←T↓X)⍳'⍺']
[7]    R←¯1↓X[⍳T←(X←T↓X)⍳'⍺']
[8]    →(∨/∧/[1](¯1+⍳⍴C)⌽(C←¯1↓X[⍳T←(X←T↓X)⍳'⍺'])∘.=N)/RIGHT
[9]    ⍞←F←(H,0⍴SC[⍴SC]←SC[⍴SC]-1),'  '
[10]   →(∨/∧/[1](¯1+⍳⍴C)⌽C∘.=(⍴F)↓⍞)/RIGHT
[11]   'THE CORRECT ANSWER IS: ';C,0⍴SC[⍴SC]←SC[⍴SC]-1
[12]   →S
[13]  RIGHT:R
[14]   →S
[15]  TRY:⍞←F←'COME ON, TRY SOMETHING!
[16]   →A
[17]  END:'THANK YOU, ',NA[⍳NA⍳' '];'  THAT''S ALL FOR NOW.'
    ∇
      ∇SETUP[⎕]∇
    ∇ SETUP D
[1]    SC←1⍴0.25×+/D='⍺'
[2]    NAME←''
[3]    NR←1⍴0
    ∇
      ∇SCORES[⎕]∇
    ∇ SCORES;C;SCR;T
[1]    C←0
[2]    SC[1];' QUESTIONS, ';T←((⍴SC)-1)÷SC[1];' STUDENTS.
[3]    'CLASS AVERAGE: ',50×+/(1↓SC)÷(⍴SC)-1;⎕AV[173]
[4]   S:''
[5]    'STUDENT:            ';NAME[(+/C↑NR)+⍳NR[1+C←C+1]]
[6]    'QUESTION NO.     ';⍳SC[1]
[7]    'POINTS (2=MAX):';SCR←SC[(⍳SC[1])+1+(C-1)×SC[1]]
[8]    'AVERAGE:            ';50×+/SCR÷SC[1];⎕AV[173]
[9]    →(C≠T)/S
    ∇
```

OF THE USER....BY THE USER....FOR THE USER

Robert L. DeMichiell
Frank Greenwood

United States Coast Guard Academy

ABSTRACT: There may have been several unspectacular successes associated with administrative data processing. But for every journalistic exposé of good news, there are multiple exposés of bad news. The purpose of this article is to show that the application of the common techniques of users groups and liaison personnel can produce some tangible benefits, even for the environment with rather unique needs. User involvement was sought and acquired as an information system was developed and implemented with limited personnel, equipment and time.

The literature abounds with computer-assisted information systems designed for the user community. Many of these systems are preceded by expressions of anticipation, such as:

> I sure hope that computer group doesn't try to help me do my job - everything they've touched so far, they've messed up.
>
> I don't like to read computer print-outs--especially the fifth carbon copy. Especially when they're always late.
>
> How do I resist the movement to automate? I've heard it costs more and employs more people. More people, more problems.
>
> I wonder what the boss thinks about information systems.

Computers have had a bad press, and one reason is that many projects concerned with the development of computer-based information systems have failed. Some have been abandoned with incurred multi-million dollar costs, few benefits and a very hesitant and cautious future clientele. The reasons usually can be traced to one or more factors: the user does not know what he needs; the computer technician does not know what the user needs; the communication gap between the two is a wide one; and, perhaps, neither reasonable expectations (costs, time lag) nor responsibilities on the part of both user and technician were clarified at the outset.

There have been unspectacular successes. Most of these have remained, as most good news does, in the background. Some computer industries and service bureaus still make money and still obligate a large share of their resources to research. Some users have chosen the right tool for use by the right people in all the right ways. Certainly, some historical knowledge about the successes and failures should shed light on the decisions of today and tomorrow.

User Involvement: Required

The purpose of this paper is to present an example of a system of the user, by the user and for the user. The pitfalls previously mentioned were heeded and some positive actions were taken to involve the administrative user, or manager, at an early stage of development of his information system (perhaps even computer-based). The dichotomy of the computing center's information system and the users of that system was avoided.

The posture was one of establishing realistic goals with progress reports instead of expected completion dates, of promoting team effort for institutional objectives rather than parochial interests, of organizing for institutional benefits rather than departmental empires, and of recognizing that people make systems work, not machines. People can make clerical systems work where automated systems fail. Even if the automated system is more appropriate.

But is this so unique? What conditions exist at the educational institution depicted here, so as to permit a healthy working relationship among the levels of management, the levels of systems development and the levels of data processing operations?

What are some of these developed systems and how did they evolve?

A Unique Setting With Unique Systems

The educational institution to be used as the example here is one of the federally-sponsored service academies, the United States Coast Guard Academy in New London, Connecticut. The enrollment is undergraduate only, and numbers approximately 1100. There are nine major fields of study with a heavy emphasis in the scientific/technical areas (mathematical, physical, ocean and engineering sciences). In addition to the mental demands placed upon the Cadet Corps by the four-year curriculum, there are requirements in the military and extra-curricular spheres as well. Physical fitness is encouraged by mandatory participation in inter-collegiate and/or intermural athletics; discipline is fostered by restrictions imposed throughout each day of 365 days each year. All cadets go to all classes; during the summer session, they are involved in programs oriented to the Coast Guard service (naval operations, cruises, mission area programs throughout the United States, etc.)

The objectives of the institution are clearly defined. The self-development of the cadet is primary and focuses on those particular traits and characteristics that merit the special trust of an officer's commission: a high sense of moral values, personal honor, integrity and a sense of responsibility. The Academy wants to graduate young people who have the ability to think clearly, critically and analytically, coupled with the perseverance to work a problem through to its solution. In addition, the developmental process must include--with equivalent priority--specific maritime and military skills such as naval operations and tactics and leadership capabilities. The aspect of leadership and its relationship to administrative skills are important in view of the normal execution of future roles in the service, some of which require a high degree of endurance and resistance to stress (particularly in search and rescue operations).

There is a five-year obligation in the Coast Guard for the cadets when they graduate. Approximately one-half of those who enter graduate. Approximately 80% remain in the service beyond the five-year period. The faculty is composed of civilians, military officers who are permanently assigned to the Academy, and rotating military officers. The permanent faculty receive pay, get promoted and aspire to tenure in much the same way as most college faculty do. The institution is governed by the military contingent.

All of the above means that there is a focus on several aspects of student development which is different from the peer population. The faculty mix is a unique feature, even with regard to the other academies. There is a sense of dedication to the institution and to the Coast Guard service--a factor which plays a large part in the purpose of an Academy to provide career officers.

This focus on several aspects of cadet development requires a strict monitoring of all activities. The program starts a year before the entering class arrives in July. The Admissions procedures have to select the most qualified applicants for appointment. Several academic and non-academic factors must be considered and, since the process is a competitive one with neither geographical quotas nor congressional involvement, it is quite complex. Cadets receive pay and allowances to defray all expenses. Tuition and room are free. Financial administration, therefore, has requirements of accountability different from a civilian college. The issuance of uniforms, summer cruise budgeting, and compatability with government procurement policies are examples of these differences. Cadets lead, and are led, as part of the training program. The brigade consists of four classes of cadets with several levels of responsibility; this occurs during the academic year as well as during the summer cruise programs. Ratings of cadets by cadets are collected on several performance criteria. Ratings by the officers corps on non-academic factors are also collected. Conduct is monitored by the cadets themselves and the faculty (military and civilian) and as such, the demerit system requires daily accounting.

There are no computer-based programs "on-the-shelf" for these specific applications. For example, in the admissions area, what college or university:

a. has to rank-order its qualified applicants so that it can offer appointments to the top 380?

b. holds a special board of officers for each qualified applicant (academically), so as to include in the overall rank-ordering other non-academic attributes?

c. **is** accountable for such a selection process to every taxpayer in the U.S.?

d. collects extensive psychological and demographic data on all entering students for use in research and counseling?

e. is required to incorporate rather detailed information on the medical status of each applicant throughout the qualifying process?

Similar examples in the other areas of concern could be cited to correspond to the special demands placed upon cadets, and therefore, upon the support team: the administrative offices in their decision-making and accountability roles.

A Viable Alternative Solution

The applicants have to be kept informed on their status (probable appointment or not) so that their possible option of attending CGA is kept alive. From an institutional viewpoint, the Academy wants the best-qualified candidates to remain in the competition. The cadets have to be kept informed of their status throughout the four-year period on all educational and military requirements.

The Academy also wants to make sure that the requirements and tasks are specified; then, performance can be evaluated according to some established standards. A quick look at the uniqueness of the overall program suggests that the manual method of data gathering and information dissemination is not appropriate. How then, can the problems be identified and any appropriate computer solutions be realized for such a given situation?

Put the analysis and programing teams to work, the user might say. Design a data base and evaluate the outputs so that reports for all the administrative offices can be derived from centralized file structures. Establish the procedures for automation and document them according to standards, so that the next person who has my job can understand it. All of these are reasonable solutions in our computerized society, even in spite of the fears and declarations of doom mentioned at the outset. The important question is: To whom are these questions addressed? More than likely the recipient is the computer/information group.

In this particular case, the group has been a rather modest staff of computer science faculty, unit-record personnel, and cadets. The equipment is second-generation batch and in the process of being updated. The high-level managers pledge support, and the users anticipate some involvement. The key issue was identified earlier as "user involvement". But how does one get the user involved?

User Involvement: Sought

The success of any information system development and implementation rests on the expectations, attitudes and performances of people. People provide organizational structures, establish the roles and responsibilities for everyone within those structures and, therefore, have control over the degree of involvement of each element. The computer people have to be involved because they push the buttons, initiate the interviews and document all that is said and done. But data are not information. Without information, there is no institution. Without the interviews and dialogue between the opposing camps (the users and the computer technicians), there is no coordination. The necessary mediating environment (users groups in this case) was created by the college president to provide an open forum of discussion of the issues by top-level managers. So the immediate image of computing activities was one of applicability of the present and projected resources to all users. It was also obvious that the computing resources and the "information system" were to be as meaningful as they wanted them to be. In other words, it is their system and each academician, administrator or researcher must be involved if they desire to compete for services. Their outputs, or summary information reports, will depend on their inputs, or contributions of time and resources.

This Computing Center Advisory Board of top-level managers establishes the policies on computing activities; computer personnel bear the burden of completing the staff work necessary for the evaluation process and for the reporting of progress. The majority opinion has a focus, and the opportunity exists for the defended minority opinion. The president assesses the situations and acts on the documented issues. Another users group of middle managers also was formed to face and resolve the day-to-day processing problems. The organizational alignment of the Computing Center, with such an arrangement as this, does not become the issue. User involvement is sought, and for those users who do their homework and are alert to the capabilities of automation, they can be rewarded.

A policy statement then was issued from the president. This document removed some of the fuzziness from the objectives of the institution with regard to computer applications. Long-range and short-term directions were stated. There was no guesswork concerning the opinion of top-level management.

The systems were developed from the "bottom-up" rather than the "top-down". One system at a time was analyzed for possible automation. The present resources had to be efficient, because they were limited. The faculty in computer science became the development team, because it was recognized by the institution as professional growth. The cadets helped in development, because it was done as computer laboratory work for regular courses for credit. All student participation was screened thoroughly so that the projects were necessary to the institution and appropriate learning experiences for the cadet. This program certainly has been well received by the cadets over the years.

In order to focus even more on the user-technician interface, computing development personnel were assigned to certain users. In this manner, the "liaison" personnel could be knowledgeable about the real requirements and needs of the user with the added advantage that each user would have to learn something about data processing as it pertains to his/her job. Thus, the realization of large amounts of expended time by both user and technician surfaced.

To suppose that such an environment was created and has existed for the past few years without its moments of frustration, arguments and personality clashes would be quite idealistic and incorrect. However, some things were done right with the help of those who failed in the past. Favorable results can result from unfavorable odds. There are tradeoffs, as long as one recognizes them.

a. The payoff is not immediate; one has to wait perhaps years to achieve the defined reasonable expectations.

b. It costs time and money to automate.

c. It requires frequent monitoring; it may detract from other operational commitments.

d. Students graduate; it requires detailed documentation.

e. Faculty applicants are limited; not all faculty are interested in combining computer science instruction with practicum.

Results and Conclusions

Approximately three hundred computer programs, most of which are quite long, were written to place "on-line" the systems for the registrar, admissions, commandant of cadets (cadet brigade matters), comptroller and other administrative elements in direct support of cadet activities. The unique requirements for each of these elements, as described in an earlier section, coupled with the fact that military officers serve four tours of duty in key positions in the organization, have required a dependence on documentation and cooperation.

The cooperative aspect was given impetus by high-level support. More important than a stated instutional policy, however, was the recognition of the efforts by the various staff elements. Several performance awards were delivered to personnel in the administrative offices as well as the computer group. Realizable goals were formulated and the milestones were accentuated with tangible monetary rewards.

The Admissions Office now can monitor over 10,000 inquiries for cadetship and keep track of the status of each of these young people throughout the formal application process. The resultant appointee would have been screened quite thoroughly for his medical, academic and non-academic credentials so that he could receive fair treatment in competition for one of the 380 positions in the entering class.

The cadets are scheduled and sectioned in a very constrained program, with little time available for any make-up work (there is no summer academic program and two-thirds of the coursework is required of all cadets, regardless of major field). Grades are reported in timely fashion. They are in the mail within two weeks of the end of the term. The academic grade point averages are important in other ways; military performance is meshed with academic performance to provide information on the "whole man" for constant evaluation for retention. Programs have been written to search the student files for cases to be evaluated.

Cadet pay and accounts operate from a developed credit card system, where the paper transactions monitor all the debits and credits, and are reported to the cadets on a monthly basis. The frequent leadership ratings by officers and cadets on cadets are automated. This data and other military performance ratings are considered when brigade and regimental positions are assigned.

Someone once said that if an institution has bad management, a computer will make it worse. Others have said that computerization would even make good management worse. For better or worse, this institution has pursued the goal of user involvement with dedication and direction and has produced some tangible benefits.

EVALUATION OF THE USE OF CAI MATERIALS IN AN INTRODUCTORY COMPUTER SCIENCE COURSE[1]

Richard G. Montanelli, Jr.
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois

ABSTRACT: A total of about 480 students in an introductory computer programming course were randomly assigned to PLATO (one hour per week using PLATO CAI lessons to learn FORTRAN replaced an ordinary lecture) and non-PLATO sections at 9:00 and 10:00 am. Results indicated that PLATO students would strongly recommend that their fellow students choose PLATO sections in the future, while students in the regular lectures were neutral. There were no differences between groups on exam or machine problem scores. There was an higher drop rate in the PLATO sections, but this was quite possibly due to factors other than the PLATO materials themselves.

I. INTRODUCTION

For the past three and one-half years the Department of Computer Science at the University of Illinois at Urbana-Champaign has been working toward automating a fraction of its teaching load (about 2,000 students per semester) in introductory computer programming courses, by using CAI materials (Nievergelt, 1975). Although separate courses are taught for different groups of students (engineers, architects, business majors, and social scientists, for example), almost all students learn FORTRAN, so CAI materials to teach FORTRAN could be beneficial to almost all students. All of these courses are taught in the same format, with two large lectures and one discussion section (15-25 students with a teaching assistant) per week. So, initially it was decided to replace one lecture per week with one hour of instruction in FORTRAN (or related programming topics) using PLATO (Alpert and Bitzer, 1970) an extremely flexible CAI system with graphics terminals.

The overall design of the PLATO materials was accomplished by the author, although most individual lessons (units of material, each designed to take 25-50 minutes of student time at the terminal) were initially written by graduate and undergraduate students as course and independent study projects under the author's supervision, and later revised and improved by members of the staff (Montanelli, 1975).[2] Thus, overall the lessons do not follow a particular teaching method or style, but each covers its assigned material in its own way. (An advantage of this strategy for lesson development is that various teaching methods are attempted, and students essentially choose the better lessons. On the other hand, an entire lesson might have to be rewritten, if students disliked it or learned little from it.) The lessons include displays of all types: simulations, graphs, diagrams, animations, interactive exercises, and text. The students interact with the lessons, taking the material at their own pace, reviewing whenever necessary and are typically quizzed during the lesson so that they can assess their level of understanding.

After several rounds of debugging and testing with individual students, and after voluntary use by 10 students during the summer of 1974 (followed by more modifications), it was felt that many of the lessons were ready for actual classroom use that fall. Accordingly, one-half of the approximately 50 students in CS 103 (an introductory computer programming class for students in the social sciences) were randomly assigned to use PLATO to replace one of their lectures. The major results of this experiment were that although students enjoyed using PLATO, there was some indication that they did not learn as much as the non-PLATO group. More specifically, the non-PLATO group had higher means on all three exams, although there were no differences in scores on computer programming assignments. The multivariate test for differences between group means was not significant ($F = 1.54$ with 4 and 39 degrees of freedom) at the .05 level. However since all the differences were in the same direction, and since two of the univariate Fs were nearly significant, the possibility does exist that the PLATO lessons were less effective than the lectures. An additional finding was that there were no differences in drop rate between the two groups. (For more details, see Montanelli, 1975).

---

[1] This research was supported in part by the National Science Foundation under Grant No. US-NSF-EC-41511. The author would like to thank Richard Anderson for help with the data analysis.

[2] The author would especially like to thank Dr. H. G. Friedman, Jr., Sandra Leach, and Jeffrey Barber for their invaluable help in this area.

Two possible major reasons for the above result were obvious to those involved in the course. First, ECS (swapping memory) for the PLATO system was at such a premium during the semester that students could only be guaranteed access to their PLATO lessons at the assigned time each week. Thus one major advantage of CAI, the ability for students to work at their own times and to review whenever necessary, was lost. (Slow students probably could not even go at their own pace, because they only had the one hour period to finish.) Secondly, many lesson inadequacies and errors were discovered. Apparently, a developer of CAI material can get useful feedback from students only if they are required to learn the material from the lessons they are studying. Otherwise, one never discovers lesson faults.

The first problem was essentially solved in January, 1975, when the amount of ECS was doubled. Additionally, during the fall and between semesters, more lesson improvements were made. This extra programming effort essentially destroyed all this author's a priori notions about how much time and effort would have to go into lesson development and improvement, before lessons would be in good shape for routine classroom use.

Thus by spring 1975, the system was ready for a large scale test. Accordingly, the approximately 600 students in CS 105 (the required introductory course for business majors) were scheduled to use PLATO twice a week. One of these sessions was a 50 minute period replacing a lecture (as in CS 103). In the second session (40 minutes), students wrote small FORTRAN programs using a specially designed, interactive compiler with excellent and extensive error diagnostics at compile and execution time (Davis, Tindall, and Wilcox, 1975). These programming sessions replaced some of the time students had previously spent doing computer programs in batch mode on an IBM 360, in the pre-PLATO version of CS 105.

Unfortunately, the additional ECS which was to be installed between semesters, arrrived late. It was installed just before the semester began and hardware problems caused PLATO to be up and down like a yo-yo during the first 10-12 days of the semester. The result of this problem was 600 irate students, many of whom were unable to successfully complete a single session on PLATO in the first two weeks, even though they were scheduled for four sessions, and many of the students attempted several makeups.

Although there was an unusually high percentage of drops during the first weeks of the semester, by the end there was evidence that the remaining students were satisfied with PLATO (54% of those completing a survey strongly recommended that a friend take a PLATO section, while only 16% recommended avoiding PLATO. Steinberg, 1975). Thus it was decided to attempt to assess student learning by administering a final exam which was constructed as a parallel form to the one used during the previous semester's CS 105 class. The exam was constructed by making minor changes (but in every part of every problem) in the old exam, such as changing constants, variable names, programming problems to similar ones (e.g., finding the minimum instead of the maximum), etc. It was felt that students would not have any advantage over the previous semester's non-PLATO CS 105 students for several reasons: a) no one in the Computer Science Department had ever (to this author's knowledge) constructed a parallel form of an old final exam, for use in a current course; b) final exams are not returned, and the previous semester's exams were safely stored away; c) only the three professors involved in the course knew of these plans; and d) virtually every answer was different from the earlier exam. The final exam results showed no large or statistically significant differences between CS 105 student's final exam scores for the two semesters, an indication that the problems with PLATO in CS 103 the previous semester had been overcome. However, a firm conclusion could not be drawn from these results because students were not randomly assigned, some of the professors in CS 105 were different from the previous semester, and information on the final exam could have leaked out.

II. PROCEDURE

In order to allow for testing some hypotheses about the use of PLATO in our introductory computer science courses, a controlled experiment was designed for CS 105, for fall, 1975. Five lecture sections were offered, with four of them arranged in the following way. Professor A taught a PLATO section (one hour on PLATO replacing one lecture) at 9:00 am, and a non-PLATO section (two lectures, no access to PLATO) at 10:00 am. Professor B taught a non-PLATO section at 9:00 am and a PLATO section at 10:00. Professor C, who was in charge of the course, taught a fifth (PLATO) section in the afternoon, but it was not involved in the experiment. It should be noted that this author did not teach any sections, and that neither professor A nor B had ever used PLATO before the semester. An additional minor difference between the two types of sections was that the PLATO students did their first two machine problems using online PLATO FORTRAN compilers instead of the IBM 360.

Before the semester began, students' registrations for the five sections were equalized (by the computer program which makes up student schedules), so that there were equal numbers of students in each section. Then students in the 9:00 and 10:00 sections were randomly reassigned to either a PLATO or non-PLATO section. Students were kept separate by using the following mechanisms. 1) Students were not allowed to transfer sections, unless it could be done without affecting the experiment. For example, a student could not stay at the same hour and change sections. If a student wanted to change from 9:00 to 10:00 (or vice-versa), he/she had to keep the same type of section (PLATO or non-PLATO). If a student wanted to switch from the afternoon section to a morning section, he/she was randomly assigned to a PLATO or non-PLATO section. Students were not allowed to transfer to the afternoon section. Any student who felt she/he could not abide by these rules was sent to the author who attempted to convince her/him of the value of educational experiments and of PLATO (There were about ten such students, and most wanted out of PLATO, presumably due to having heard about the problems from the previous semester). The result of these discussions was that two students who protested violently against PLATO and machines in general were allowed to transfer out of PLATO sections. 2) Students in non-PLATO sections were not given access to PLATO, as a student record with associated name and code had to be created for each PLATO student, and this was not done for non-PLATO students; 3) Individual attendance was taken in the lectures intended for non-PLATO students only, insuring that no PLATO students could enter. In order to assess the effect of these precautions, students were asked (via a questionnaire given to all students immediately after completing their CS 105 final exams) if they had looked at materials for the other group (they were assured that there would be no penalty). A few non-PLATO students had seen some of the PLATO materials through friends or through having access through other courses, and a few PLATO students attended lectures, either through using the name of a friend who was not attending or through slipups in our records. It was felt that none of these minor disturbances would have any major effect on results.

The three hypotheses of this study were:

1. PLATO students would enjoy the course more, and give it a stronger recommendation to their friends.
2. PLATO and non-PLATO students would perform equally well on exams and homeworks in the course.
3. The drop rates in the two types of sections would be similar.

III. RESULTS

In answer to the question (from the questionnaire administered with the final exam): 'If a friend were taking CS 105 next spring and PLATO and non-PLATO sections were offered, what would you recommend he take?' PLATO students strongly recommended PLATO (112 circled 'definitely PLATO', 88 'PLATO if convenient', 45 had 'no recommendation', 15 said 'lecture if convenient', and 21 said 'definitely lecture'. On the other hand, non-PLATO students were neutral (their responses, in order, were 29, 22, 91, 26, 19), or even showed a slight preference for PLATO.

In order to compare learning across groups, a 2x2 univariate analysis of variance was computed for each exam and for total points on computer programs. No significant differences were found, and means were nearly identical for the various groups (Table 1).

The third hypothesis concerning drop rates was rejected, however. Professor A had 19 (15%) drops from his PLATO section, and only 4 (4%) from non-PLATO. Professor B had 28 (25%) drops from PLATO, and 18 (14%) from non-PLATO.

Table 1
Average Scores on Exams
and Machine Problems

			PLATO	non-PLATO
(a) Machine Problems				
	Professor	A	143	142
		B	140	146
(b) Hour Exam 1				
	Professor	A	92	91
		B	91	91
(c) Hour Exam 2				
	Professor	A	65	62
		B	62	62
(d) Final Exam				
	Professor	A	140	134
		B	135	134

IV. DISCUSSION

Students in the PLATO groups would strongly recommend that their friends choose PLATO sections, thus confirming the first hypothesis. Even if the 'extra' 25 drops in the PLATO sections were strongly negative, they would have a small effect on the totals of 200 PLATO students recommending PLATO, and only 36 of them recommending lectures. It should be remarked that when the PLATO students were asked to indicate what they thought were the worst features of PLATO, 78 checked 'The distance to CERL' (Unfortunately the terminals are located on the north edge of campus in CERL, about a mile from most commerce courses.), while 37 checked 'Lack of human contact', and 31 checked 'PLATO going down', the next two most frequently checked responses. Thus the major problem was unfortunately out of our control.

The second hypothesis was not rejected. The nearly identical scores on exams and machine problems in Table 1 are evidence of that, without reference to statistics. There is no reason to suspect that the PLATO drops were poor students. However, if the dropped PLATO students were below average, they could not have had a large enough effect on the results to alter the obvious conclusion. This result is certainly in agreement with most studies of the effects of CAI. In fact, when Jamison, Suppes, and Wells (1974) surveyed the effectiveness of alternative instructional media, they stated:

> '... the equal-effectiveness conclusion seems to be broadly correct for most alternate methods of instruction at the college level ...',

and suggested studying costs of various methods of delivery. However, a major advantage of CAI is that once used, it is not set in stone like a textbook or movie. As a result of this experiment, the two lessons which students liked the least are being rewritten from scratch. Secondly, a quiz system has been begun. When completed, it will present a quiz to each student at the completion of each lesson. The quizzes are not written by the authors of the lessons, and in fact quiz authors are discouraged from looking at the lessons. However, the quizzes are written from the same objectives that were used to write the lessons. The resulting quiz scores will not only tell the students how well they understand the material which the lesson is supposed to cover, but will tell instructors and lesson authors how well the lesson is working. Thus, continual improvement is possible, and perhaps eventually CAI materials will be as good as the best lecturer, and therefore better than many.

On the other hand, the hypothesis about equal drop rates was rejected. This was a surprising result, especially when the smaller experiment a year earlier (under worse conditions) showed no differences. However, the earlier course may have been a special case. It was a relatively small, elective course with mainly juniors and seniors in psychology and similar fields. These students were more involved and interested in the experiment (as the author was teaching the course), and they may have stayed for that reason. On the other hand, CS 105 is a required course for freshmen in the college of commerce, and the students were presumably less interested in long term educational goals (for themselves as well as for the PLATO materials). However, although this drop rate was disturbing, there were a few, likely reasons for it, all of which could be fixed. For one thing, the first three weeks were confusing for the students because they had pre-enrolled in a course which they expected would consist of two lectures and a discussion each week. Instead, three-fifths of them had a lecture cancelled and had to sign up for a PLATO section instead. These sections caused a lot of trouble, as some were scheduled for weekends, and many students complained that they were unable to meet any of the remaining available PLATO times. Although most of this confusion was necessary due to the nature of the experiment, in the future students will preregister for PLATO sections just as for any other class. A second possible cause for the different drop rates was that for the first few weeks, PLATO students were required to do their programming problems in one of the online, interactive compilers. Although it was thought that this would be fun for the students, the compiler gave very poor response time because of the amount of processing going on to check for errors after each student keypress. Finally, drops might have been due in part to student dissatisfaction with the two poor lessons which were later rewritten. Students had not been systematically polled about the lessons before, and the relatively negative reaction to two of them was quite surprising.

Another possible explanation for the higher drop rate on PLATO, is that some students (< 10%) are anti-machine and that CAI will always have this problem. The author does not feel that the large differences found here could be attributed to this reason. However, this is being checked during the current semester, because the problems mentioned in the preceding paragraph have been fixed, and although no experiment is being run during the current semester (all five sections of CS 105 are using PLATO), the current drop rate could be compared with past drop rates, in order to see if the PLATO drop rate is as high as was found here.

V. SUMMARY

PLATO lessons can be used to replace one lecture a week in an introductory computer programming course. Students learn as much and prefer PLATO to large lecture sections. The remaining problems are: 1) Is there a higher drop rate on PLATO? and 2) Can instruction be improved through continued development of the CAI materials?

References

Alpert, D. and Bitzer, D. L. Advances in Computer-based Education. Science, 1970, 167, pp 1582-1590.

Davis, A., Tindall, M. H. and Wilcox, T. R. Interactive error diagnostics for an instructional programming system. Proc. Fifth Symp. on Computer Science Education, SIGCSE Bulletin, 1975, Vol. 7, No. 1.

Jamison, Suppes, and Wells. The Effectiveness of Alternative Instructional Media: A Survey. Review of Educational Research, Vol. 44, No. 1, Winter 1974, pp 42-67.

Montanelli, R. G., Jr. CS 103 PLATO Experiment, Fall 1974. DCS Report No. 746, University of Illinois at Urbana-Champaign, July, 1975.

Nievergelt, J. Interactive Systems for Education - The New Look at CAI. Invited paper. IFIP World Conference on Computer Education, Marseilles, France, September, 1975.

Steinberg, E. R. PEER Evaluation Report, CS 105. Unpublished report, Computer-based Education Research Lab., University of Illinois at Urbana-Champaign, May 1975.

THE VIDEO-DISC IN AN EDUCATIONAL ENVIRONMENT

By

Demetrios A. Michalopoulos
Computer Science Programs
California State University, Fullerton

1. Introduction

One of the challenging fields of computer science is the computer graphics field which, because of its relatively high cost, has not been used extensively for educational purposes. Recent hardware developments in the graphics technology and price breakthroughs in digital equipment are beginning to influence the educational uses of computer graphics. If "a picture is worth a thousand words" and if the cost is not at this ratio, one can provide the student, not necessarily of computer sciences, with the possibility of a continuous inter-action in which the student can see the results of his decision and thus allows him to experiment and practice with ideas not completely understood yet.

This paper presents a design for an interactive educational system incorporating the recent development of the video-disc.

2. Background

Computer graphics is a fairly recent addition to computer systems and an even more recent addition to educational type systems. Ten years ago the idea of a computer controlled interactive graphics system in education was a revolutionary one. However, today with the existing progress in hardware and the continuing price break-through in computer systems one realizes that the idea has come a long way but it still has a long way to go. It appears at this time that computers will be as indispensable in education as they have been in practically every other field. The typical computer educational system comprises of a display unit, local storage of sorts, a controller for the display, a means of interaction with the system usually a keyboard, and an intelligent part which could be either a local processor or a time sharing computer.

Different systems approach the above from diverse points of view. The Plato terminal, for example, is comprised of a plasma display unit with a computer controlled rear projection microfisce unit. The terminal communicates typically over the telephone with the computer. The Ticcit system comprises of a small color television set modified to display computer generated characters with high resolution, plus a special typewriter keyboard. Audio and video for the terminal is generated by a local minicomputer.

The typical proposed Video-disc player oriented system is composed of a commercially available television set (preferably a color one), a video-disc player unit interfaced to a local processor and a means of interaction with the system. The local processor approach is not an exclusive one, conceivably one could go over the telephone.

In order that the advantages of the Video Disc be more clearly understood the next section describes in short the principles of its operation.

3. Principles of Operation

The video-disc is being developed presently by at least half a dozen companies with another half a dozen doing some work on it. The major target market being the home entertainment one. Their projected price for the player is less than \$500 and less than \$10 for a 30-60 minute disc. These prices are within economic reach by practically everyone owning a color television set and thus the market is a very wide one. Because of the above reason the educational applications (especially with digital interactive and storage capabilities) are sort of pushed down the stack.

Conceptually, the players are split into two major categories: (a) Contact Video Disc systems, (b) Contactless Video Disc systems; for the passive i.e., narrative, lecturelike type of educational system both types are well suited. However, for the interactive type of education the contact video-disc is fairly useless. This is because the contact type uses a stylus to track the disc and therefore wear is introduced like the conventional audio records.

The contactless type of video-disc player, on the other hand, uses some type of light to track the disc, laser light in some, to conceivably white flood light to others. At any rate, there is no physical contact with the disc and, therefore, no wear. However, the biggest advantage is that we can introduce now, very desirable features like single framemovement at variable speed, freeze-frame, as well as slow motion and fast motion both in forward or reverse. The player delivers a US-NTSC 525 line compatible color signal video format modulated on a low rf carrier to any standard television receiver.

The contactless video-disc players are basically two different systems. Firstly described is the Phillips-MCA system which is comprised of a low power laser light focused into varying sized oblong depressions. The light beam's reflected light is lost because it is diffracted over angles larger than the microscope objective is capable of accepting. This type of modulation is detected by a photodiode and is electronically processed into sound and picture. The second type is being developed by i/o metrics. The disc is basically a photograph of direct analog recording of the video signal on a track of varying width and transparency, omitting any carrier frequency.

The basic advantages of the video-disc are summarized as follows: (a) High Information Density (b) Random Accessibility (contactless type).

4. Description of Operation of the System

This section describes the principles of operation for the proposed video-disc based interactive educational system. Two designs are presented, one that assumes a number of features not presently available and one that could be implemented with the advertised (existing features.)

The first type of design assumes certain features which although not available today on existing players are certainly within the reach of existing technology. One can envision the video-disc being separated into two types of "files" or "frames". One type for audio and video type (TV) signals and the other for digital type data. With the above assumption, not an unreasonable one, the operation of the system follows:

The digital information storage "frames" are containing code, in ready to execute format, for the local processor preferably a microprocessor or a minicomputer. On command by the system this code is read into the local storage, control is then passed by the local operating system to the program just read in which in turn controls the following sequence of video frames according to the answers for given questions asked by the system (written on the TV screen, or I/O device). When the section of the code in the system has been executed it causes the video-disc player to position itself over the digitally encoded frame containing the next part of the control program or the previous part depending on the validity of the returned answers. The interaction could be either through a keyboard or through a cursor or a "touch panel".

Due to the built-in capability of the player of fast forward, fast reverse and random accessibility of the frames a student could start at the beginning or from a particular section of the programmed material. This process could be accomplished either by selecting the "frame" through a menu or by manually overring the system by selecting the corresponding frame through the player directly. The software structure required in order to accomplish the proposed method is fairly straight forward.

5. Block Design

This section describes two possible configurations for the video-disc educational system incorporating a local mini or micro. The basic configuration is presented in figure 1 and consists of the video-disc player and controller, the host computer, some local memory, a color monitor and a keyboard input-output device.

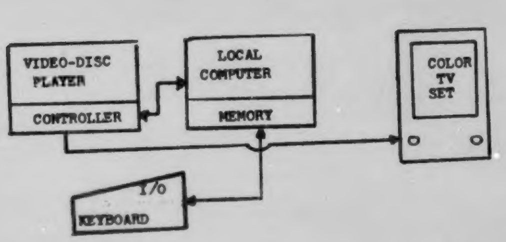


Figure 1. Basic System

The basic system could be expanded to become a more "human factor" oriented system with the addition of the following capabilities. The expanded system is presented in figure 2 and it consists of the following additions, a graphics controller with local memory, thus, enabling the user to produce computer generated displays (overlays) text editing and drawing compositions and in general enhance the man/machine communication. A "touch panel" could also enhance the educational potential of such a system.

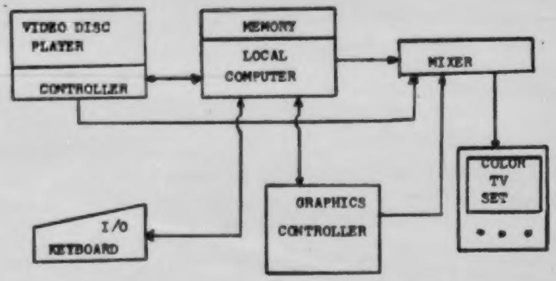


Figure 2. System with graphics capability.

6. An Alternate Design

An alternate design that is viable with the existing form of the video-disc is presented in this section, i.e., no digital storage only TV audio and video information.

a. Software Structure. The control program for each part the programmed material resides in some mass storage device i.e., floppy disc or cassette. Resident in the memory there is a table with the correspondence between frames being displayed and control code. Assuming that the local memory cannot contain the complete control program some algorithm could be developed that could be used to load from the mass storage device the control code in conjunction with the correspondence table and the answers given by the user.

b. Hardware Structure. This section shows the hardware configuration for a video-disc based educational system incorporating only existing video-disc features (as advertised) in conjunction with a local processor, the video-disc player, a TV color monitor, a keyboard input-output device, and a local mass storage device. The local storage could be either a cassette system or a floppy disc system.

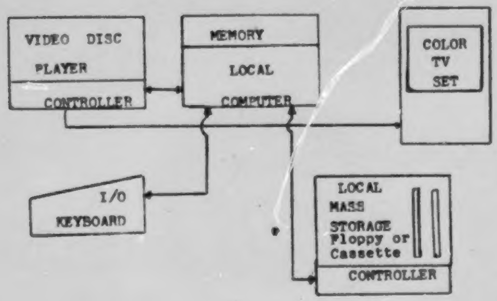


Figure 3. Alternate Block Design.

7. Conclusions

This paper has presented possible ways of incorporating a fairly new development, the video-disc, in an educational environment. The principles involved are that of the state of the art and the advantages are great,both educationally and economically. With announced prices of less than $500 for the player for the home entertainment model one could expect a computer controlled version to be in the market soon. The digital storage feature is such an important part to all data processing people using fixed data bases (let's not forget that the video-disc in its digital storage version is nothing but a giant ROM of 10^{10} bits) that is bound to be available soon. The announced price for the disc of less than $10 which, if one keeps in mind that they last forever (almost), makes a very cost-effective approach to storage of fixed data bases (or at least ones that do not require constant update). Finally the system in its expanded form lets the student (user) and the computer operate in a way that is optimum for both.

8. References

1. K.D. Broadbent, "A review of the MCA Discovision system" presented at the 115th SMPTE Technical Conference, April 26, 1974, Los Angeles, California.

2. J.A. Jerome, "Film-Based Video Disc System" presented at November, 1974 meeting of the SPIE, Northern California Chapter.

3. D. Mennie, "Television on a Silver Platter" IEEE Spectrum, August, 1975, pp. 34-39.

4. A. Korpel, "A Review of the Video Disc Principles" SID Intrl Symposium, May, 21-23, 1974, San Diego, California.

SIPS - A STATISTICAL INTERACTIVE PROGRAMMING SYSTEM

Edwin L. Anderson

School of Education, Oregon State University

ABSTRACT: SIPS, an interactive command processing system, was developed at Oregon State University to provide interactive program library support for statistical instruction and for data analysis. The uses of the various subsystems of SIPS (REGRESS, ANOVA, MONTE CARLO, MANOVA, MULTIVARIATE, ENUMERATE) range from simple computations to the teaching of elementary methods of data exploration and model building in the style that is possible with interactive computing. This paper presents a brief non-technical overview of SIPS.

1. INTRODUCTION

The Statistical Interactive Programming System (SIPS) was developed for a time-sharing environment at Oregon State University in a cooperative effort on the part of the Department of Statistics and the OSU Computer Center. SIPS was specifically designed to provide interactive program library support for statistics instruction. The development of the system started in 1969 and has been continually revised and added to during the past 6 years.

During this time period, SIPS has been increasingly used by faculty members for statistical instruction in various disciplines. As more faculty members have become acquainted with SIPS and its potential, the trend has been towards using SIPS as a major tool for data analysis of individual research projects. At this point in time, there are as many calls per month for research purposes (1500) as for instructional purposes (1500). Faculty members and graduate students have expressed satisfaction with the rapid interactive response time and the convenience and flexibility of the system. In addition, the elimination of numerous and/or intermediate output has been pleasing and less confusing to the majority of the users. As such, it has detracted from the use of an excellent collection of conversational and batch-made single-purpose programs in the Computer Center Library.

A short statement from the SIPS User's Manual (1) is quoted to inform the reader of the value of an interactive system.

> Interactive use grants the computer user a unique freedom--he need not completely specify the entire sequence of steps his analysis is to consist of; he may examine intermediate results, then decide to modify his future steps or his data. To a statistician this freedom is especially important, for to truly understand a data collection he must be permitted to examine it readily in a variety of ways. Often the exact steps his analysis should take cannot be predicted in advance. In that case he may either demand excessive computations or leave some aspects of his problem unexamined.

2. OSU COMPUTING ENVIRONMENT

SIPS was developed under and operates most effectively under the OS-3 (Oregon State Open Shop Operating System) system. Development of the OS-3 system was accomplished primarily by a group of undergraduate students during the mid and late 60's while working on a generalized command processing systems project. OS-3 runs on a CDC 3300 computer equipped with executive hardware and virtual memory paging capability. The executive program is written in the assembler and is therefore tied to the CDC 3300 and not easily transportable.

Development of SIPS on the OS-3 system has virtually stopped at this point because computer center personnel are now almost totally involved with the new CYBER 70 - Model 73 and its KRONOS operating system. SPSS and BMD are available on the CYBER but not in an interactive mode. Consideration is being given to a new and transportable system of SIPS on a dedicated minicomputer or the CYBER. The latter would probably begin by complementing SIPS through the development of a rudimentary executive to work with large storage requirements and the heavier computing of regression and multivariate analysis.

Under OS-3, user programs (such as SIPS) are allocated a maximum of 65,000 24-bit words of memory and access to the central processor in one hundred millisecond time slices. OS-3, like other virtual memory time-sharing systems, employs demand-paging techniques to effectively utilize computer memory space. Demand-paging allows a portion of the user's program area to be absent from computer memory and, instead, stored on some external mass storage device. When the computer detects a reference to a non-resident area of the user's program an interruption is generated to notify the operating system to transfer that section into computer memory for execution. Much of the SIPS internal design concepts were employed to function in cooperation with the demand-paging algorithm.

CDC 3300 Configuration

Core memory	98,304 24 BIT words (1.6 microsecond cycle time)
Tape drives (4)	150 inches/second (200,556, or 800 BITS/inch density)
Disk drives (5)	8,192,000 characters/drive
Mass storage disk file	268,435,456 characters (100 milliseconds max seek time)
CRT displays (16)	20-line screen, 50 characters/line
Remote terminals (150)	teletypes and Tektronix graphics

CYBER Configuration

Core memory	65,000 60-BIT words (1 microsecond cycle time)
Tape drives (4)	75 inches/second (200, 556, or 800 BITS/inch density)
Disk drives (2)	117,000,000 characters/drive
Remote terminals	teletypes (64)

Each system has its own card reader, card punch and line-printer. Two X-Y plotters are shared via mag tape - CALCOMP and GERBER flatbed. A PDP-8 serves as a satellite to the CDC 3300. A PLATO terminal is available for demonstrations only. On the average there are 39,346 jobs per month on the 3300 and 6,387 jobs per month on the CYBER. The average number of users at any one time is approximately 40.

3. USE OF SIPS

The instructional use of SIPS has progressed from one class in statistical methods within the Statistics Department to every class (except Operations Research and Advanced Mathematical Statistics) within that department. Within many schools of the university some use is made of SIPS in an instructional mode. The use runs from a purely computational tool to the teaching of elementary methods of data exploration and model building. As such, it has taken over a large part of the entire statistical computing load of the university. Within the School of Education, SIPS is used for several courses such as Educational Measurement, Research Procedures, Applied Statistical Design, Applied Statistical Procedures, and Introduction to Computers for Teachers.

As more students have become involved through instruction, faculty members have been exposed to SIPS and its applications in data analysis. After initial involvement with the system, faculty members have been using SIPS more often for their own personal use in analyzing data. To assist faculty members in becoming familiar with the system, the Computer Center and the Department of Statistics sponsor numerous workshops and short courses. Within the School of Education, several faculty groups have become familiar with SIPS and its potential through in-service meetings. Manuals have been developed to instruct the uninformed but interested parties.

There is a great variety in the level of statistical complexity at which SIPS is applied by users. Since SIPS originated as a teaching tool, it retains much of the character of that origin. All use starts by learning to manage data (read data, modify data, enter data, etc. . .). Learning progresses through various levels of statistical training from graphical and arithmetic description to simple testing of one and two variables, and on to Chi-square, regression, and multivariate analysis, depending on the course objectives. The level of knowledge of the course instructor becomes an exceedingly important factor in determining whether the interactive capability is really used or SIPS is used only as a computational device.

Those beginning to use SIPS as a research tool must also learn the rudimentary elements of the system and data management techniques. In most instances, the interactive capabilities of SIPS will not be utilized to a great extent until the user becomes quite familiar with the system and is encouraged to take advantage of one or more of the major subsystems.

4. GENERAL COMMAND FORMAT

As a terminal-oriented, interactive command processing system designed to support instruction and educational research, SIPS commands are simple, brief, and of a logical nature to the user. The commands are in the form of a keyword and, if needed, parametric information to define specific variables, observations, or optional procedures. The keyword logically reflects the function to be performed. Abbreviations may be used in some cases but are generally discouraged. Basically, the user enters a command to the system through a terminal keyboard (batch-processing is also available) and receives statistical results or other information in response. Responses from the system are normally brief and highly readable. The length of the response is dependent on the type of analysis requested. Extraneous and voluminous outputs are usually not necessary nor desirable when output is being produced at a terminal. However, the user may give all commands from a terminal and have the responses printed on the line-printer. This procedure is effective in some instances but does limit the interactive capability of SIPS.

SIPS currently offers over two hundred operational commands to the user. The general form of a command is shown below. The command is in the form of a keyword which requests a specific function to be performed. As indicated above, additional information (e. g. variables to be considered) is required with many commands, and optional features are available for some. These take the form of a parameter string.

KEYWORD
or
KEYWORD, PARAMETER STRING

Examples:

LIST	lists all observations for all variables in current data file
CENTRAL, 3	requests several measures of central tendency for variable 3 (arithmetic, geometric, and harmonic means; median; quartiles; mid-range; maximum and minimum values)
CENTRAL, 1, 2, 4	same as above for variables 1, 2 and 4.
CENTRAL, 1 to 5	same as above for variables 1 to 5
CORRELATION,1,4	Pearson product-moment correlation coefficient between variables 1 and 4
CORRELATION, 1-4	same as above except gives Pearson r for all pairs of variables from 1 to 4
SCATTER, 2, 7	requests scattergram for variables 2 and 7. Variable 2 on horizontal axis versus the values of variable 7 on vertical axis.
SCATTER, 2, 7, ( )	parameter string permits user to define lower and upper bounds for each variable
REGRESS, 5, 1-4	control is transferred to the REGRESS subsystem with the first variable (5) identified as the dependent variable.

5. SIPS AND SUBSYSTEMS

OS-3 is the control mode monitor for the CDC3300.

There are a number of systems available through OS-3, but the most important for the SIPS user are SIPS and EDIT. EDIT, the OS-3 text editor, is used to file new data, modify existing data, or replace old data. Data may be entered for SIPS in free-form or formatted as desired. In the latter case, the appropriate format statement must be given when entering SIPS. It is also possible to enter data directly to SIPS from a terminal. This procedure should be used only if the number of variables and the number of observations per variable are limited in size. Generally, it is more efficient to create a data file in the EDIT mode and have the data read directly into SIPS.

Keywords

ABORTOFF
ABORTION
APPEND
AUTOCORR
AVERAGE
C
CENTRAL
COEFVAR
COMMENT
COMPARE
COMPRESS
COMPUTE
CONTAB
CORRELATION
COVARIANCE
CROSSPROD
DESCRIBE
DLC
DO
DURBIN-WATSON
EJECT
ENUMERATE
EQUIP
ERASE
EXIT
FAULTS
FORMAT
FORMIN
FORMOUT
FREQ
FREQHIST
FTEST
GEOMEAN
GRANDMEAN
HISTOGRAM
IF
INSTRUCT
IQRANGE
KDLC
KFORMAT
KFORMIN
KFORMOUT
KRUSKAL-WALLIS
KURTOSIS
LIST
LOG
MANN-WHITNEY
MAX
MEAN
MEDIAN
MIDRANGE
MIN
MONTE-CARLO
MULTIVARIATE
N
NAME
NAMELIST
NEWS
NMISSING
NOFAULTS
NOMINAL
NORMALPLOT
NUMERIC
ONEWAYAV
ORDER
OUT
PAIREDT
POOLVAR
RANGE
RANK
RANKCORR
RCROSSPROD
READ
REGRESS
REPLACE
REWIND
RMD
RSUMSQRS
SAMPSIZE
SCATTER
SERIAL
SHIFT
SIGNEDRANK
SIGNTEST
SIZE
SET
SKEWNESS
SPACE
STANDARDIZE
STATUS
STDERRM
STDEV
STOPLOG
SUM
SUMSQRS
T
TALLY
TIME
TRAFFIC
T-SQ
TSS
TSUM
TTEST
TTYOFF

TTYON
UNEQUIP
V
VAR
VARIANCE
WILCOXON
ZEROALLOW
ZERODELETE
PROTECT
UNPROTECT

In addition to the keywords listed above, there are pre-programmed functions to compute square root, natural and common logs, anti-logs, trigonometric functions, absolute values and rounding operations.

The SIPS user has access to six major subsystems described below.

REGRESS. The REGRESS subsystem is used to perform multiple regression analyses. Computations in this subsystem are centered around construction of a model--the least squares approximation of the dependent variable by a linear function of one or several independent variables. Simple commands allow the user to add variables or drop variables from the model and at each stage to request information about the current model. These commands include requests for an analysis of variance table, the multiple correlation coefficient, the partial correlation matrix, residuals, t-values and other appropriate data for model building. It is through the use of these commands that the full interactive potential of SIPS is realized. Most of the regular SIPS commands, as listed above, are also available to the user in the REGRESS and other subsystems. In addition, each subsystem has its own set of unique commands.

In using REGRESS, the ability to routinely examine residuals contributed much to the functional guidance of variable selection and to useful consideration of model adequacy. The user makes the final choice of the model based on the functional logic of the included variables and the behavior of the model for reasonable ranges of combinations of the variables.

Keywords

ADD
ADDVAR
AVTABLE
BACKSTEP
DEP
DROP
DROPVAR
END
ESTIMATE
INVERSE
MODEL
NOMODEL
NORMEQ
PRECISION
PREDICT
R-SQ
RCOEFSE
RESIDUAL
RMATRIX
STEPWISE
TRANSFORM
YHAT

In addition, when parameter strings and optional features are added to these keywords, over 40 commands are available under the REGRESS system. As an example, the keyword ADD may have the following format:

ADD, (LIST)
ADD, (LIST), F
ADD, ALL
ADD, ALL, F
ADD, BEST
ADD, BEST, F
ADD, BEST, (LIST)
ADD, BEST, (LIST), F

MULTIVARIATE. As a subsystem of SIPS, MULTIVARIATE handles the computations of multivariate analysis. Like REGRESS, this system allows interaction between computer and user at each step in the computation, and the user may revise his approach to analysis at any time. Once again, the user may call for any of the SIPS commands to assist in the analysis. Since most techniques of multivariate analysis are based on computations involving the covariance or correlation structure of the data, the system first computes and stores a complete covariance matrix for the variables designated by the user.

Keywords

PRINCOMP
CANCORR
ROTATE
VECTOR
AXIS
END

ANOVA. This subsystem provides SIPS users with a convenient and easy method of obtaining specific analysis of variance tables. The computations are based on completely balanced factorial designs. This situation is not a serious limitation because most familiar designs can be specified in this way. Using ANOVA commands one can combine several main effects and interaction mean squares (and their corresponding degrees of freedom) to yield an analysis appropriate for a split plot design, a nested design or a choice of many other special designs. Specific multiway classification means are available from a single command.

Keywords

AVTABLE
EXCLUDE
POOL
CLASSMEANS
F
RESET
DESIGN
INTERMEANS
END

MANOVA. The MANOVA subsystem may be thought of as an extension of the REGRESS system to the multivariate case. It is not currently operational at this date. When control is transferred to MANOVA, the first response is a printing of the residual sum-of-squares matrix of the p dependent variables after regressing on the q independent variables. Keywords, such as START, ADD and ROTATE are then supplied by the user to examine the discriminant functions and the linear combination of the dependent variables which provides the greatest association with the independent variables. At each stage of the analysis, the significance of the model increment is tested and statistics displayed.

MONTE-CARLO. The MONTE-CARLO subsystem is designed to simulate repeated sampling from certain populations and to accumulate as SIPS variables the observations in the samples and/or designated statistics of the samples. The user need only specify the population from which he wishes to sample, parameters of the distribution, the number of samples desired and the sample size. Having generated this data, the user may use SIPS statistical information requests, or commands from other subsystems, to perform the computations to analyze the data and to gain insight into the behavior of the sampling distribution. Thus, MONTE-CARLO is used primarily as an instructional aid.

Populations currently available

BETA BINOMIAL CHISQUARE
DISCRETE ERLAND EXP
F GAMMA GEOMETRIC
INTEGER NEGBINOMIAL NORMAL
POISSON T UNIFORM

Keywords

GENERATE INSTRUCT KEEP
POP POPULATIONS PRINT
PRINT STATISTICS SUBTRACT

ENUMERATE. The ENUMERATE subsystem provides the ability to do tabulations and compute the Chi-square statistic. It is designed to handle only discrete (counting or enumerative) variables in numerical or nominal form. Tabulation is possible for selected values only or by combining specified sets of values. Tabulations as percentages are also possible. Chi-square tests are possible for "goodness of fit" and tests of independence in two and three dimensional arrays.

There are three stages to the use of ENUMERATE.

(1) Data value grouping
(2) Tabulation
(3) Summary information

Keywords

CHISQUARE COMBINE EXCLUDE
PERCENTAGES RESET SLICE
TABULATE END

6. PROGRAM AND DATA STRUCTURES

SIPS consists of about 10,000 lines of machine language coding and over 20,000 lines of FORTRAN coding. The architectural design is that of a resident executive with support package and a group of over one hundred overlay and information files. The program coding levels are easily dichotomized as the resident section comprises all of the machine language coding while all the overlay sections are coded in FORTRAN.

The resident executive is the primary control section of the SIPS system. It initializes the user memory area, performs input and output of all data observations, handles all terminal communications, performs command and parameter processing, and executes all non-overlay and non-major subsystem commands. It is coded in re-entrant format which allows it to be shared among many OS-3 users at once. In this way the demands upon the OS-3 memory resources by SIPS users are primarily for numerical data storage and functional overlays to service the individual user's needs.

Transient routines (overlays) perform almost all major statistical and transformational functions. These FORTRAN programs are stored as OS-3 core-image overlay files. To perform the above functions, the overlay routines access the COMMON storage area and communicate with the executive program for data observation access and input/output functions.

Major subsystems are groups of programs which perform interrelated processing. Most have a matrix or data structure on which pertinent processing is done. All of them have their own executive section and common storage area. The major subsystem executive acts in cooperation with the SIPS main executive in the function of command processing. The SIPS executive performs its command processing functions normally for all commands except those relevant to the major subsystem. These commands are transferred to the major subsystem executive for further processing. The execution of these commands may involve only associated routines in the subsystem executive area, or may require an overlay to be loaded from external mass storage. In this manner, a comparatively small amount of fixed memory space can be used to execute a large number of different programs. While the transient overlay technique can be thought of as "swapping" in programs for independent execution, major subsystems can be considered as "mini" command processing systems which are swapped into memory in similar concept to the overlay programs. Major subsystems allow for more complex and flexible analysis of data.

The major data sections include (1) the data observations, (2) a COMMON area for summary storage and communications, and (3) a command table for command recognition, program initiation, and parameter string processing directives. As any OS-3 program can at most reference two banks of 32K words in each, the programs and summary information are placed in one bank (lower memory), and all data observations are placed in the other bank (upper memory).

The data observations are arranged so that all variables for a single observation are in consecutive memory locations. Since all variables may not have the same number of observations, provision is made to allow for missing observations. The data observation area is used only for storing the user's data.

Summary information for data variables, system status information, linkages between programs, variable identification, and user-entered parameter information are all available in COMMON. The summary information consists of simple statistical information for each variable. The system status information denotes such conditions as major subsystem operation, successful data entry, activity logging, control of output information, and other pertinent system conditions. The linkages between major subsystems are contained in COMMON to allow indirect access to the executive. This maintains the re-entrat format of the executive program and also allows the execution to be changed without requiring re-creation of all application overlays. Data variable identifications, input/output buffer areas, and parameter string decoding areas are also found in the COMMON area.

The command table contains a sequence of command definitions consisting of a keyword prototype, status indicators, and functional information. The command table employs two separate units: the internal table section and the external table section. The internal table is coded as part of the SIPS executive program and is partitioned as to character length of keyword and may contain commands with keywords from one to sixteen characters in length. By partitioning the commands in this manner, only the subset of commands with keyword length equal to that entered by the user need be searched. The external table is initially stored on a random access file and consists of a directory and a list of command definitions. This list of commands is divided into partitions of variable length and command groups for each major subsystem. During the initialization of the SIPS system, the first group in the external table is transferred by the random file into a memory storage area. The second group will not be accessed unless a major subsystem or a command definition is called for by the user.

All of the features of SIPS (internal structural design, commands, interactive capabilities, subsystems, ease of entering data and response time) all combine to make SIPS a very useable and efficient system.

7. SUMMARY

Although SIPS began as a tool to assist with statistical instruction, the trend has been for more users to interact with SIPS for the purpose of analyzing data. Designed and developed by a joint effort between the Computer Center and the Department of Statistics at Oregon State University, SIPS has continued to grow and become more sophisticated during the last six years. The internal design and the program and data structures were developed to use a limited amount of fixed memory space. The short and precise commands, consisting of keywords and parameter strings, were designed for the convenience of the user. The addition of major subsystems, such as REGRESS, has permitted the user to truly interact with the computer and to control the pattern or sequence of the analysis of data.

8. REFERENCES AND ACKNOWLEDGEMENTS

1. Guthrie, Donald, Carole Avery and Keith Avery. Statistical Interactive Programming System (SIPS), User's Reference Manual. (Corrected Edition - 1974). Oregon State University Bookstore, Corvallis, Or. 97331

2. Rowe, Kenneth E. 1975 News Notes For SIPS Users, Statistical Computing Report No. 2. January 1975. Department of Statistics, Oregon State University, Corvallis, Or. 97331.

3. The author of this paper would like to further acknowledge Donald Guthrie (currently with UCLA Neuro-Psychiatric Institute, Keith Avery (currently with Boeing Aerospace Company), and Kenneth Rowe (OSU) for additional information given through formal and informal presentations.

9. BIOGRAPHICAL INFORMATION

Edwin L. Anderson received degrees from the University of Kansas (1952), the University of Washington (1959) and Oregon State University (1970). After sixteen years in the public schools in positions ranging from mathematics teaching to counseling to administration, he joined the ranks of higher education. Currently an associate professor in the School of Education at Oregon State University, his major responsibilities are in teaching--Educational Psychology, Foundations of Education, Measurement and Evaluation, Research Procedures and Computer Literacy. He has used SIPS as an instructional tool and for data analysis with his own research.

GNOSIS - A SWEDISH/ENGLISH PACKAGE FOR COMPUTER ASSISTED INSTRUCTION

Shirley Chann, Susan Finch, & Clare Hamlet

Pima Community College, Tucson, Arizona

ABSTRACT: GNOSIS is a system that helps an instructor write a computerized lesson for student use on an interactive terminal. It provides a lesson, as designed by the instructor, composed of text, questions and answers (expected and unexpected). It allows a response to student answers with hints and reinforcement suggestions. Unexpected answers, student suggestions and comments are saved for aid in improving the lesson. Extensive branching capabilities are included and can be incorporated into a lesson to advance, skip, or to repeat sections. These branches can be optional or conditional (the result of a correct answer or the result of "50%" correct answers). The lessons produced become object code which executes faster and with less core than some other systems. GNOSIS is written in ALGOL for the DEC System-10 computer by Jacob Palme of the Research Institute of National Defense, Stockholm, Sweden. It is available through the DECUS Program Library.

For the last few years, some of the faculty members at Pima Community College have been using the computer as a tool to teach various subjects, including Physics, Chemistry, Sociology, Music, Math, Art, and Computer Science. The computer used is a DEC System-10 which supports many terminals on a time-sharing basis.

The Computer Science faculty members decided to augment the classroom experience with student/computer interaction. Because of the large number of students and the number of different instructors offering Introduction to Computers, it was important to develop a delivery system to insure that a common set of objectives would be learned by all students. The Computer Assisted Instruction approach provides standards for a specific level of learning without hindering the freedom of the instructor's style of teaching in the classroom. It lends itself to the open-entry, open-exit concept of education. Students may proceed as rapidly as desired, or they may repeat a set of objectives as often as needed. It also frees the instructors for more individualized teaching on a one-to-one basis. The CAI lessons are designed to reinforce the subject after review of the text materials and classroom lectures.

A grant for a pilot project was given during the summer of 1975 to investigate the available CAI packages, select one of them, and implement several lessons. The GNOSIS package was selected because it met the intent of the project and required minimum software modification for the computer system. The lessons developed during the summer were tested and modified during the Fall semester. This pilot project can serve as a model for future development of CAI lessons at Pima Community College.

Preparing a lesson for CAI, using GNOSIS, involves several activities:

I. Planning the lesson.
II. Writing the lesson using GNOSIS commands.
III. Directing the computer to generate and execute the lesson.

I. Planning the lesson:

To plan a lesson, the instructor should select the the topics to be covered, and organize them into a logical order. Usually the student has a reference text or has attended a lecture on the topics, so text type explanations are normally short summaries. This, however, can vary by topic or by the emphasis the instructor wishes to place upon it. A problem can arise in that the number of topics selected produces too much lesson for a reasonable time. This can be resolved by eliminating some topics, or by splitting the lesson into two lessons. It is important to know how the commands of the GNOSIS package work, as this will affect the format of the lesson.

Planning the questions and answers to test the students' mastery of a topic presents the greatest challenge. Subjective questions with a relatively high number of "free form" answers are possible, but VERY DIFFICULT to execute. What is easy to compose on paper does not come out as it was planned when it is tested. The subject matter may not be precise enough to limit the "free form" answers so the program can check them. Spelling, mathematics, and language offer good possibilities for this type of question and answers. (see exhibit #1) Objective questions with multiple choice or true-false type answers are much easier to handle. Since the instructor controls the format of all possible answers, those answers are all the program must check. (exhibit #2)

The evaluation of the answer provides a supplementary reinforcement for student motivation and success. The correct answer brings out a compliment and allows the student to proceed or skip (at the instructor's option) to another section. A wrong answer causes a response pointing out the incorrectness, giving a word of encouragement, and returning the student to either the related text or question for another try. Responses vary and are part of GNOSIS. If the instructor wishes to put in additional comments, these are easy to include. A neutral answer (neither right nor wrong) also can be included, with appropriate instructor remarks. It is important to allow for the "unexpected" answer, or else the program will give a "wrong" but just go on to the next section. These unexpected answers will be listed on the teacher report, and may be useful in lesson modification.

Branching is easy with GNOSIS. The instructor can ask a question and, as a result of the student's answer:

1. Terminate the lesson and tell the student to read the textbook.
2. Branch back to the beginning.
3. Return to a previous text or question.
4. Skip to a new section.

Thorough planning of the lesson, combined with a good understanding of the GNOSIS commands and how they operate, will lead to a good lesson which is easy to write and run.

II. Writing the lesson using GNOSIS commands:

Each GNOSIS command is preceded by a percent sign (%). The next % will act as the terminator for the current command.

OPTIONAL COMMANDS:

%LANGUAGE -- Allows English or Swedish to be used and defaults to English if omitted.

%TEACHER -- Produces a report for the instructor which analyzes and summarizes the students' responses. The name and address of the instructor are the only entries for the command.

MAIN COMMANDS: (see exhibit #3)

%TEXT -- One or more lines of text are written after this command.

%QUESTION -- One or more lines are entered to state the question to be answered by the student.

%RIGHT -- The correct answer is included after this command. If more than one answer is correct, a separate %RIGHT command must be used for each correct answer the instructor will accept.

%WRONG -- The author writes the anticipated incorrect answers after this command. A separate %WRONG command must be used for each incorrect answer that is expected. A %WRONG command followed by blanks should be used for any unexpected wrong answers in addition to the anticipated ones stated in the previous %WRONG commands.

GNOSIS automatically gives the student an appropriate indication that the answer is right or wrong. If further comments are to be made (e.g. addition reinforcement or clues), they should be included on the next line(s) after the %RIGHT or %WRONG command. If identical comments apply to several successive right or wrong commands, the %SAME command can be used by inserting the %SAME command instead of repeating the comments.

%END -- This command must end every lesson.

MODE COMMANDS

The following commands provide flexibility in how the students' responses will be compared to the expected right or wrong answers. This flexibility is provided in two areas:

1. Words in the student response may or may not have to be in the exact order of the answer. (see exhibit #4)

%ORDER -- After this command, student answers will be accepted only if the words are in the same order as the expected pattern.

%NOORDER -- After this command, student answers will be accepted even if the words are not in the same order as the expected pattern.

2. Extra words in the student answer, which are not found in the expected response, may be allowed or prohibited. (see exhibit #5)

%EXTRA -- Student answer will be accepted even if it contains extra words which are not part of the expected answer pattern.

%NOEXTRA -- Student answers will not be allowed to contain any extra words which are not part of the answer pattern.

The default combination is %EXTRA and %NOORDER. A mode command remains in effect until it is changed.

BRANCHING COMMANDS:

To adjust the lesson to the students' achievements, a student may be allowed to skip parts of the lesson, and another who needs remedial work can branch to it.

A label is required to branch to a different point within the lesson. Labels can be assigned to either %TEXT or %QUESTION commands by including them immediately after the command and following the label with a colon. Preassigned standard labels can also be used to:

1. Repeat the last question.
2. Go to the next text or question.
3. Go to the beginning of the lesson.
4. Go to the end of the lesson.

%GOTO -- The label given in this command indicates where to continue in the lesson from this point. (see exhibit #6)

An example of GNOSIS lesson incorporating the previous commands can be found in exhibit #7.

III. Directing the computer to generate and execute the lesson:

After the lesson has been written on paper, it can be converted into computer readable form by punching the lesson statements on to punched cards, or typing directly into the computer using an interactive terminal and any text editor available on the system. The lesson should be verified to catch keypunching or typing errors.

Once the lesson is stored on a secondary storage device, following steps are needed to transform the lesson into executable form for the student: (see exhibit #8)

1. The GNOSIS program will read the lesson, edit the lesson to catch syntax errors, and if any, print out appropriate error messages so that the user can correct the lesson. When the lesson does not contain syntax errors, GNOSIS generates ALGOL source program as output.
2. The ALGOL compiler is called in to translate the ALGOL source program into a machine language program.
3. The machine language program is loaded into the memory of the computer and saved on disk.

At this point, the students can use a terminal to call for the lesson desired. The computer follows the logic of the lesson, prints the texts and questions, accepts answers, analyzes the answers and gives appropriate right or wrong reinforcements, until the logical end of the lesson, or interrupted by the student.

Upon completion, a teacher report may be printed showing the number of questions, number of right and wrong answers, a list of the unexpected answers, along with comments from the student to the instructor.

One approach to learning how to write a GNOSIS lesson is:

1. Plan a whole lesson on paper.
2. Handle one topic, with it's questions and answers before going on to the next.
3. Enter and test the first topic thoroughly (it may be necessary to restructure the questions and answers).
4. Be sure all possible answers are covered.
5. Have someone else use the lesson to catch any errors or ambiguous questions.

After the first lesson is implemented, the next will be much easier. Knowledge of the text, question and answer patterns help in planning and writing a second lesson.

Advantages of GNOSIS lessons, as with other CAI methods can be:

1. The active participation of the student encourages motivation and learning.
2. Immediate reinforcement for right answers, and guidance to correct wrong responses help the student evaluate his performance.
3. The lesson can be adjusted to the student to allow for individual differences and competencies.

At Pima Community College, the Computer Science faculty is continuing to develop GNOSIS materials and to evaluate their effectiveness as a supplement to classroom instruction. As the school grows, and more faculty become interested in CAI, a workshop to assist other faculty members to implement their own GNOSIS lessons is an immediate goal of the Computer Science Department staff.

EXHIBIT #1

```
%QUESTION
WHICH DAYS COME BEFORE WEDNESDAY IN A WEEK?
%RIGHT MONDAY TUESDAY
%WRONG MONDAY
YOU ONLY GAVE ONE OF THE TWO DAYS BEFORE WEDNESDAY.
%WRONG TUESDAY
%SAME
%WRONG
A HINT:  THERE ARE TWO DAYS BEFORE WEDNESDAY.
%WRONG
A FURTHER HINT:  THE DAYS AFTER WEDNESDAY ARE THURSDAY FRIDAY
SATURDAY AND SUNDAY.
%END
```

```
WHICH DAYS COME BEFORE WEDNESDAY IN A WEEK?
?MONDAY TUSDAY
Wrong!
A HINT:  THERE ARE TWO DAYS BEFORE WEDNESDAY.
Try again.

WHICH DAYS COME BEFORE WEDNESDAY IN A WEEK?
?MONDAY
No!
YOU ONLY GAVE ONE OF THE TWO DAYS BEFORE WEDNESDAY.
Try again.

WHICH DAYS COME BEFORE WEDNESDAY IN A WEEK?
?THURSDAY
No!
A FURTHER HINT:  THE DAYS AFTER WEDNESDAY ARE THURSDAY FRIDAY
SATURDAY AND SUNDAY.
Try again.

WHICH DAYS COME BEFORE WEDNESDAY IN A WEEK?
?MONDAY AND TUESDAY
Well, there's hope!
The teacher was not expecting this part of your answer:
AND
```

EXHIBIT #2

```
ALL RIGHT THEN - LET'S REVIEW A LITTLE.
THE FEATURES OF A DIRECT ACCESS FILE INCLUDE:
A.  THE ABILITY TO GO DIRECTLY TO THE RECORD WANTED
    WITHOUT GOING THROUGH THE ENTIRE FILE.
B.  SPEED IN FINDING AND UPDATING A SMALL AMOUNT
    OF RECORDS IN A LARGE FILE.
C.  THE ABILITY TO REWRITE AN UPDATED RECORD IN PLACE,
    RATHER THAN WRITING IT IN A NEW PLACE
    AND REMOVING THE OLD RECORD.
D.  THE POSSIBILITY OF USING A FILE SEQUENTIALLY OR
    DIRECTLY, BY ORGANIZING IT WITH INDEXES.
                 To continue push RETURN.


WHICH OF THE ABOVE FEATURES IS DIFFERENT FROM
THE FEATURES OF A TAPE FILE?  PLEASE ANSWER A, B, C, D,
E (ALL OF THE ABOVE) OR F (NONE OF THE ABOVE).
?A
Wrong!
NOT TOO BAD, BUT HOW ABOUT A BETTER ONE?
Try again.


WHICH OF THE ABOVE FEATURES IS DIFFERENT FROM
THE FEATURES OF A TAPE FILE?  PLEASE ANSWER A, B, C, D,
E (ALL OF THE ABOVE) OR F (NONE OF THE ABOVE).
?F
Wrong!
COME ON NOW.  HAVE YOU REALLY STUDIED THIS CHAPTER?
REMEMBER THAT TAPE FILES ARE SEQUENTIAL AND MUST BE USED
SEQUENTIALLY.  THIS MEANS NOTHING CAN BE DONE
WITH DIRECT ACCESS, THE WHOLE FILE MUST BE PROCESSED,
AND IT IS DANGEROUS TO TRY TO UPDATE IN PLACE.



WHICH OF THE ABOVE FEATURES IS DIFFERENT FROM
THE FEATURES OF A TAPE FILE?  PLEASE ANSWER A, B, C, D,
E (ALL OF THE ABOVE) OR F (NONE OF THE ABOVE).
?G
No!
COME ON, ANSWER ONE OF THE LETTERS YOU ARE ASKED FOR.
                 To continue push RETURN.


WHICH OF THE ABOVE FEATURES IS DIFFERENT FROM
THE FEATURES OF A TAPE FILE?  PLEASE ANSWER A, B, C, D,
E (ALL OF THE ABOVE) OR F (NONE OF THE ABOVE).
?E
Very good.
```

EXHIBIT #3

```
%TEXT
THERE IS A RIVER WHICH DIVIDES THE UNITED STATES INTO EAST AND WEST SECT
IONS. THERE IS A STATE BY THE SAME NAME.
%QUESTION
WHAT IS THE NAME OF THIS RIVER AND/OR STATE?
%RIGHT MISSISSIPPI
%RIGHT MISISSIPPI
%RIGHT MISSISIPPI
%RIGHT MISSISSIPI
%WRONG COLORADO
%WRONG MISSOURI
%WRONG OHIO
%WRONG
%END
```

```
THERE IS A RIVER WHICH DIVIDES THE UNITED STATES INTO EAST AND WEST SECT
IONS. THERE IS A STATE BY THE SAME NAME.
                   To continue push RETURN.


WHAT IS THE NAME OF THIS RIVER AND/OR STATE?
?COLORADO
Wrong!
Try again.

WHAT IS THE NAME OF THIS RIVER AND/OR STATE?
?OHIO
No!
Try again.

WHAT IS THE NAME OF THIS RIVER AND/OR STATE?
?MISISSIPPI
Now you're catching on!
```

EXHIBIT #4

```
%TEXT
SOME PHRASES MAKE SENSE IN ANY ORDER THE WORDS MAY BE ARRANGED
%QUESTION
WHAT IS A COMMON ITALIAN DISH THAT MOST CHILDREN LOVE
%RIGHT SPAGHETTI AND MEATBALLS
%WRONG
%TEXT
OTHER PHRASES NEED TO BE IN AN EXACT ORDER TO MAKE SENSE
%QUESTION
WHAT PHRASE IS OFTEN USED TO REFER TO LOVELY COMPLEXION
%ORDER
%RIGHT PEACHES AND CREAM
%WRONG
%END
```

```
SOME PHRASES MAKE SENSE IN ANY ORDER THE WORDS MAY BE ARRANGED
                 To continue push RETURN.


WHAT IS A COMMON ITALIAN DISH THAT MOST CHILDREN LOVE
?MEATBALLS AND SPAGHETTI
Good for you!



OTHER PHRASES NEED TO BE IN AN EXACT ORDER TO MAKE SENSE
                 To continue push RETURN.


WHAT PHRASE IS OFTEN USED TO REFER TO LOVELY COMPLEXION
?CREAM AND PEACHES
No!
Try again.

WHAT PHRASE IS OFTEN USED TO REFER TO LOVELY COMPLEXION
?:PEACHES AND CREAM
Now you're catching on!
The teacher was not expecting this part of your answer:
:
```

EXHIBIT #5

```
%TEXT
EXTRA WORDS ARE ALLOWED IN THE STUDENT ANSWERS IN SOME CASES
%QUESTION
IN WHAT CONTINENT DO WE LIVE?
%EXTRA
%RIGHT NORTH AMERICA
%WRONG
%TEXT
SOME QUESTIONS DO NOT ALLOW THE STUDENT TO USE EXTRA WORDS
%QUESTION
COMPLETE THE FOLLOWING ADDRESS WITH THE NAME OF A STATE
1500 S. ELM, TUCSON,---------
%NOEXTRA
%RIGHT ARIZONA
%WRONG
%END
```

```
EXTRA WORDS ARE ALLOWED IN THE STUDENT ANSWERS IN SOME CASES
                 To continue push RETURN.


IN WHAT CONTINENT DO WE LIVE?
?AMERICA
Wrong!
Try again.

IN WHAT CONTINENT DO WE LIVE?
?WE LIVE IN NORTH AMERICA
Now you're catching on!
The teacher was not expecting this part of your answer:
WE LIVE IN




SOME QUESTIONS DO NOT ALLOW THE STUDENT TO USE EXTRA WORDS
                 To continue push RETURN.


COMPLETE THE FOLLOWING ADDRESS WITH THE NAME OF A STATE
1500 S. ELM, TUCSON,---------
?IT IS ARIZONA
No!
Try again.

COMPLETE THE FOLLOWING ADDRESS WITH THE NAME OF A STATE
1500 S. ELM, TUCSON,---------
?ARIZONA
Well, there's hope!
```

EXHIBIT #6

```
%TEXT LBL1:
%QUESTION
WHAT NUMBER FOLLOWS 6?
%RIGHT 7
%GOTO LBL3
%WRONG
%QUESTION
WHAT NUMBER FOLLOWS 3?
%RIGHT 4
%WRONG
%GOTO LBL1
%QUESTION LBL3:
WHAT NUMBER FOLLOWS 101?
%RIGHT 102
%WRONG
%END
```

```
WHAT NUMBER FOLLOWS 6?
?8
Wrong!
Try again.

WHAT NUMBER FOLLOWS 6?
?9
No!
Correct answer:
7

WHAT NUMBER FOLLOWS 3?
?5
No!
WHAT NUMBER FOLLOWS 6?
?8
No, you're wrong this time.
Try again.

WHAT NUMBER FOLLOWS 6?
?9
Not quite.
Correct answer:
7

WHAT NUMBER FOLLOWS 3?
?4
You're doing fine.
                         To continue push RETURN.


WHAT NUMBER FOLLOWS 101?
?102
Exactly right!
```

EXHIBIT #7 (page 1)

```
%LANGUAGE ENGLISH
%TEACHER CSC100, UNIT 4
%TEXT
THE BASIC PHILOSOPHY OF PUNCHED-CARD DATA PROCESSING IS
TO RECORD SOURCE-DATA AS HOLES PUNCHED INTO CARDS AND TO
PROCESS THIS DATA WITH THE USE OF MACHINES.
%ORDER
%QUESTION
WHAT TWO CHARACTERISTICS ALLOW PUNCHED-CARD DATA PROCESSING
TO BE AN ADVANTAGEOUS METHOD OF PROCESSING DATA?
  (GIVE THE CORRECT NUMBER FOR YOUR ANSWER.)
   1  MACHINE SPEEDS AND REPLACING CLERKS
   2  REPETITIVE USE OF DATA AND INTEREST IN WORKING WITH MACHINES
   3  TO UTILIZE KEYPUNCH TALENTS AND THE HOLLERITH CODE
   4  REPETITIVE USE OF DATA AND MACHINE SPEEDS
%RIGHT  4
%WRONG 1
REPLACING CLERKS IS NOT A PRIMARY ADVANTAGE OF PUNCHED CARD DATA
PROCESSING. REREAD THE TEXT AND THINK HARDER.
%WRONG 2
INTEREST IN WORKING WITH MACHINES IS NOT AN ADVANTAGE OF THE
PUNCHED CARD METHOD FOR PROCESSING DATA.
%WRONG 3
UTILIZING KEYPUNCH TALENTS WOULD NOT BE NECESSARY IF THERE WERE
NOT ADVANTAGES TO THE PUNCHED CARD METHOD OF DATA PROCESSING.
WHAT ARE THE ADVANTAGES?
%WRONG
YOU MUST CHOOSE AN ANSWER: 1,2,3 OR 4
%QUESTION QUES10:
HOW MANY ROWS ARE THERE ON A STANDARD PUNCHED CARD?
%RIGHT    12
%RIGHT TWELVE
%WRONG   13
CLOSE, BUT THINK OF THE ROWS CAREFULLY AND TRY AGAIN
%WRONG    11
%SAME
%WRONG  80
NO, THAT IS THE NUMBER OF COLUMNS
%WRONG
%QUESTION
WHICH ROW IN THE CARD CAN BE USED AS EITHER A ZONE OR DIGIT CODE?
%RIGHT  0
%RIGHT ZERO
%WRONG
THINK OF THE ZONES (12,11,AND 0) AND THE DIGITS (0-9)
%QUESTION
WHAT WOULD BE COMBINED WITH A 0- ZONE TO CONSTRUCT THE
LETTER "W" IN THE HOLLERITH CODING SCHEME?
%RIGHT   6
%WRONG 5
 YOU FORGOT THAT THE S-Z GROUP OF LETTERS STARTS WITH THE 0-2
 INSTEAD OF 0-1 ZONE/DIGIT COMBINATION
 %WRONG
 REMEMBER THAT THE GROUP OF LETTERS A-I HAS A HOLLERITH
 12-ZONE; J-R HAS AN 11-ZONE, S-Z HAS A 0 ZONE
```

EXHIBIT #7 (page 2)

```
%QUESTION
WHAT LETTER IS REPRESENTED BY THE ZONE-11 AND THE DIGIT-4
%RIGHT  M
%GOTO QUES60,
%WRONG
%TEXT TXT10:
SINCE YOU SEEM TO NEED MORE REVIEW ON HOLLERITH HERE IS ANOTHER
QUESTION
%QUESTION
WHAT CHARACTER IS REPRESENTED BY THE 11-ZONE AND THE 8-DIGIT
%RIGHT Q
%WRONG
%GOTO REPEAT
%QUESTION QUES60:
WHAT IS THE NAME GIVEN TO A GROUP OF COLUMNS ON A
CARD WHICH REPRESENTS ONE DATA ITEM?
%RIGHT  FIELD
%RIGHT FEILD
THE CORRECT SPELLING IS FIELD...BUT YOUR ANSWER IS ACCEPTABLE
%WRONG
REMEMBER THAT THE HOLES PUNCHED IN CARDS IS TO RECORD DATA, AND
THIS DATA IS BROKEN UP INTO MEANINGFUL GROUPS. WHAT ARE THEY CALLED?
%QUESTION
HOW MANY TIERS ARE THERE IN THE PUNCH PORTION OF THE 96 COLUMN CARD?
%RIGHT   3
%WRONG
REMEMBER THE 96 COLUMN CARD HAS 32 POSITIONS IN EACH PUNCH ROW
%TEXT
REMEMBER THE 96-COLUMN CARD HAS A CODING SCHEME OF 6 ROWS PER
CHARACTER (OR PER COLUMN)..B,A,8,4,2,1
%QUESTION
WHAT CHARACTER IS REPRESENTED IN THE 96-COLUMN CARD BY THE B-ZONE
AND 4,2 DIGIT CODE
%RIGHT O
THIS WAS AN EXCELLENT ANSWER.  THE 6-BIT CODING SCHEME WILL BE
COVERED MUCH MORE THOROUGHLY IN CHAPTER 7
%RIGHT 0
YOU GAVE THE NUMERIC 0 INSTEAD OF THE ALPHA O, BUT IT'S OK
%WRONG
THE ANSWER SHOULD BE AN ALPHABETIC O CHARACTER BUT
THAT IS OK SINCE THE 6-BIT CODING SCHEME ISN'T OFFICIALLY COVERED
UNTIL CHAPTER 7
%GOTO NEXT
%QUESTION QUES100:
TO SORT AN ITEM NUMBER FIELD IN COLUMNS 1-20 USING A SORTER, WHICH
COLUMN WOULD BE SORTED FIRST
%RIGHT 20
%WRONG
REMEMBER THE ORDER OF ANY SORT FOR A MACHINE IS FROM LEAST SIGNIFI-
CANT TO MOST SIGNIFICANT
%QUESTION
TO SORT A DECK OF CARDS BY CITY WITHIN COUNTY WITHIN STATE, WHICH
OF THE FIELDS WOULD BE THE LAST TO BE SORTED?
%RIGHT STATE
%WRONG
THE MOST SIGNIFICANT FIELD MUST BE SORTED LAST
```

EXHIBIT #7 (page 3)

```
%QUESTION IF SCORE < 4 THEN GOTO TXT50:
DO YOU FEEL THAT YOU SHOULD REPEAT THIS LESSON?  (YES OR NO)
%NEUTRAL YES
%GOTO START
%NEUTRAL NO
%GOTO TXT60;
%QUESTION TXT50:
YOU DID NOT ANSWER ENOUGH QUESTIONS CORRECTLY THE FIRST ATTEMPT
I AM SENDING YOU BACK TO REPEAT THE LESSON
TO CONTINUE PUSH RETURN.
%NEUTRAL
%GOTO START
%TEXT TXT60:
ALL RIGHT, SINCE YOU ANSWERED THE QUESTIONS WELL ENOUGH
%END
```

EXHIBIT #8 (page 1)

```
.RUN GNOSIS[600,611]

GNOSIS version 1.1 March 1974.
Input name of lesson file with extension or push return to stop.
UNIT4.GNO
Translation of the lesson 'UNIT4.GNO' has begun...
...and is ready.
Input name of lesson file with extension or push return to stop.


END OF EXECUTION - 10K CORE

EXECUTION TIME: 22.05 SECS.

ELAPSED TIME:   2 MINS. 28.33 SECS.

.COMPILE UNIT4.ALG
ALGOL: UNIT4

EXIT

.LOAD UNIT4
LINK:   Loading

EXIT

.SAVE UNIT4
Job saved

.RUN UNIT4

This is the lesson 'UNIT4' produced by the GNOSIS system
for computer aided instruction(CAI) from the Research
Institute of National Defence, Stockholm, Sweden.
The teacher who wrote this lesson is:
CSC100, UNIT 4

IMPORTANT: You can interrupt the lesson wherever you want by typing
the single character '%'
on your terminal and then pushing the RETURN button.
Thereafter, you can
- leave the whole lesson or
- skip just one question or
- back up to a previous part of the lesson.
```

EXHIBIT #8 (page 2)

```
THE BASIC PHILOSOPHY OF PUNCHED-CARD DATA PROCESSING IS
TO RECORD SOURCE DATA AS HOLES PUNCHED INTO CARDS AND TO
PROCESS THIS DATA WITH THE USE OF MACHINES.
                  To continue push RETURN.


WHAT TWO CHARACTISTICS ALLOW PUNCHED-CARD DATA PROCESSING
TO BE AN ADVANTAGEOUS METHOD OF PROCESSING DATA?
   (GIVE THE CORRECT NUMBER FOR YOUR ANSWER.)
   1  MACHINE SPEEDS AND REPLACING CLERKS
   2  REPETITIVE USE OF DATA AND INTEREST IN WORKING WITH MACHINES
   3  TO UTILIZE KEYPUNCH TALENTS AND THE HOLLERITH CODE
   4  REPETITIVE USE OF DATA AND MACHINE SPEEDS
?2
Wrong!
INTEREST IN WORKING WITH MACHINES IS NOT AN ADVANTAGE OF THE
PUNCHED CARD METHOD FOR PROCESSING DATA.
Try again.

WHAT TWO CHARACTISTICS ALLOW PUNCHED-CARD DATA PROCESSING
TO BE AN ADVANTAGEOUS METHOD OF PROCESSING DATA?
   (GIVE THE CORRECT NUMBER FOR YOUR ANSWER.)
   1  MACHINE SPEEDS AND REPLACING CLERKS
   2  REPETITIVE USE OF DATA AND INTEREST IN WORKING WITH MACHINES
   3  TO UTILIZE KEYPUNCH TALENTS AND THE HOLLERITH CODE
   4  REPETITIVE USE OF DATA AND MACHINE SPEEDS
?4
Now you're catching on!
                  To continue push RETURN.%
Answer 'STOP' if you want to leave this lesson,
'SKIP' if you want to skip this question,
'BACK' if you want to go back to an earlier part of the lesson.
'CONTINUE' if you want to continue.
?STOP

End of lesson 'UNIT4'.
You have answered   1 questions, and you gave the right answer   1 times
.
On   0 questions your answer was right the first time you got the questi
on.

This lesson has produced a report sheet for the teacher.
This sheet will be printed on the line printer.
Please send the line printer listing (not the listing
at the conversational terminal!) to the teacher.
He needs the listing to improve the lesson.
You can send it anonymously - the teacher will
not know which student sent it to him.

The name and address of the teacher is:
CSC100, UNIT 4


                  To continue push RETURN.

If you have any message for the teacher, then write it now.
Finish with an extra push on the RETURN key.

?
Exiting from lesson 'UNIT4'.
```

CMI WITH A NEW TWIST

Lawrence P. Huelsman, Lotus M. Knief, and John Luiten

University of Arizona, Tucson, Arizona 85721

ABSTRACT: On the basis of improved performance with the use of tutorial CBI lessons, a highly flexible and machine-independent system for an individualized CMI course is being developed. In addition to optional lessons, the system will include a router to access lesson, test, problem application, and record files. A unique feature is the application package with diagnostic feedback.

The project to be described in this session represents the further development of an individualized computer-based instruction system already in operation at the University of Arizona, called LERN. One application has been used with 70 tutorial statistics lessons developed by Knief, which have markedly improved student performance in various classes in several departments.

A multiple regression analysis of Educational Psychology statistics students using and not using the lessons was calculated. Results revealed a difference between means, significant at the .01 level, in favor of those using the lessons, with neither sex nor class standing (i.e., whether a student was junior, senior, or graduate) having any effect. Furthermore, the semester grade distribution in classes using the lessons has changed from the typical normal shape to a skewed distribution with more A's and B's. Along with the markedly improved performance has been a very favorable acceptance and appreciation of the lessons by the students.

The LERN system, however, lacks several desirable features to make it even more widely applicable: (1) a system for completely individualizing instruction for large numbers of students; (2) an individualized problem application and feedback package; and (3) an aid for prospective instructors. In the past, attempts to individualize learning have included such efforts as creating multiple, smaller-sized sections of a large class; hiring numerous instructors to teach these sections; making use of tutors, help sessions, hand-out sheets of problems; and other such devices--all of which have not solved two major problems: 1) individualization of large numbers of students on a widespread basis; 2) achieving such individualization in the face of currently severe hiring and budget restrictions.

There is a general agreement that the use of computers in education can enhance the efficiency and effectiveness of learning. The reason apparently, is the provision of improved learning conditions especially in relation to the computer's ability to create a highly interactive learning environment where learning can be individualized (Bell, 1974; Educational Technology, Nov., 1974, entire issue).

Even though the computer has been found to be effective for learning in a context of individualized instruction, those educators most in need of its assistance often are the least prepared and able, in terms of time and skill, to develop large amounts of computer-based instruction. These educators also tend to actively fear computer technology, which adds to their disinclination to use the computer's capability in their own instruction.

What is needed is a managed individualized instruction system to help solve these problems on a widespread basis and at reasonable cost in keeping with today's budgetary constraints. This system would provide for pre- and post-testing of students, presentation of assignments, administration of individualized application problems with a unique step-by-step diagnostic and remediation capability; with the option of adding computer-based instruction lessons if desired. The system should be as easy as possible for instructors to use, to overcome their fears and help insure adoption.

Computer-managed instruction (CMI) appears to be one of the best solutions for the following reasons (Baker, 1971; Belt & Skubal, 1974; Lippey, 1975; Spuck & Owen, 1974; Young, 1974):

1. It uses conventional materials and operates well within the traditional classroom setting, reducing costs and major educational program alterations.
2. It solves the most serious problem of individualization for large numbers of students, i.e., management.
3. It is nonthreatening to instructors who fear computers, since it is a "tool" for their assistance in managing individualized classes.
4. Since it is controlled by the teacher, it has greater probability of being used.
5. It helps ease the staffing and budgetary difficulties in education, because one or two instructors with an assistant or two

could handle large numbers of students in an individualized course using CMI.

The goal then is to develop a universally transferable system (written in FORTRAN) for individualizing instruction for large numbers of students with minimal cost, use of few faculty, and use of ordinary instructional materials.

Basically, the system would consist of four parts:

1. A Router;
2. Computer-based instruction Lessons;
3. An innovative step-by-step individualized Problem Application and Feedback Package;
4. A Monitor.

The Router would access and manage test item, feedback, assignment, problem, and performance files for student interaction. The interaction would be of the following type:

1. A pre-test.
2. If criterion is met, the post-test could be taken immediately for grade.
3. If the criterion is not met, the learner would be given an assignment, possibly computer lessons, and application problems for study and feedback before the post-test could be taken for grade.
4. The system would randomly generate items for the pre- and post-tests and for the application problems.

The Lessons provide a tutorial mode. A complete set of such lessons already are available for statistics students; their success was described earlier.

The Problem Application and Feedback Package truly is unique and is described below.

The interactive course preparation Monitor would be used by instructors at a terminal. The monitor would lead an instructor painlessly through the steps necessary (entering items, assignments, answer decision rules, problems and feedback) to use the system.

Such a complete system as outlined could be used in any course at any educational level. The completeness of the system and relative ease of use should make adoption more probable and the potential cost-benefit ratio quite attractive.

Development of the Problem Application and Feedback Package is a truly unique innovation in that it would:

1. generate an individualized data set for each student to use with problems applying the concepts he has acquired in his lessons. Each student's data set would be different from the data set provided for any other.
2. increase the development of the student's problem solving ability by diagnosing his results on a step-by-step basis and by giving feedback and providing remedial learning as necessary at each step.
3. record the progress of each student, correcting his work as necessary, and providing him with additional assignments as indicated on the basis of performance.

The advantages of providing such a step-by-step capability to augment the learning provided by the system are many. Some of the most significant ones are:

1. the student would be able to get immediate assistance when he encounters a difficulty in his problem solving efforts. Further, this assistance would be available at any time, day or night, without being constrained to the availability of an instructor.
2. the help that would be available for the student would be on a personal basis similar to a one-to-one relationship, which is not the case in the typical class.
3. the student's work would be evaluated and graded immediately after submission, since he would receive immediate feedback on the correctness of his work, and, if necessary, be given remedial learning to correct his errors.
4. the student would be compelled to work individually since his data set would be different from that of any other student.
5. since each student would need to apply what he has learned, it would make that learning relevant. This means that each individual student would transfer, something that has been extremely difficult to insure in ordinary instruction.

As may be seen, this project has a good lesson basis already developed and being used actively. Further documentation of the use of the lessons will become available as a result of research being conducted currently. A number of time series and regression studies are being done on use factors such as computer time per lesson and its variation, and lesson use relative to time of day, week, and month. These factors are being related to lesson performance and course performance.

It may be seen that we are attempting to make significant additions to an ongoing, successful system to establish it as a completely viable system with the distinctive features described. The additions promise to be of direct practical value, to be widely applicable, and to result in more effective and cost-efficient instruction for each student. The savings in multiple faculty salaries, extra help sessions, tutoring, and hours of time for both faculty and students are incalculable but considerable. Furthermore, this system makes the typically exorbitant costs of materials development unnecessary, since an instructor's own exercises, test items, assignments, and so on may be utilized. We believe that the proposed system translates into an increase in student-teacher ratios and a decrease in learning time, all with improved learning, and thus into lower over-all costs.

References

Baker, F. B. Computer based instructional management systems: A first look. Review of Educational Research, 1971, 41, 1, 51-70.

Bell, F. H. Why is computer-related learning so successful? Educational Technology, 1974, 14, 12, 15-18.

Belt, S. L. & Skubal, J. Computer managed instruction: An application. AEDS Journal, 1974, 8, 1, 24-32.

Educational Technology, 14, 11. Nov., 1974, entire issue.

Lippey, G. Computer managed instruction: Some strategic considerations. Educational Technology, 1975, 15, 1, 9-13.

Spuck, D. W. & Owen, S. P. Computer managed instruction: A model. AEDS Journal, 1974, 8, 1, 17-23.

Young, K. C. Using a computer to help implement the Keller method of instruction. Educational Technology, 1974, 14, 10, 53-55.

COMPUTER AIDED INSTRUCTION IN SIERRA VISTA (ARIZONA) ELEMENTARY SCHOOLS

Julia T. Feugate
Title I Math Consultant
Sierra Vista Public Schools, Sierra Vista, Arizona 85635

ABSTRACT: A commercially available Computer Aided Instruction (CAI) system is briefly described. Its application in the elementary grades of a small school system (Sierra Vista Public Schools) is discussed. Student performance before and after introduction of CAI is analyzed to assess impact of the system. Some of the factors to be weighed in selecting CAI systems are presented. Suggestions are given for establishing a successful CAI progr[illegible]

This afternoon I'm going to discuss several aspects of the Computer Aided Instruction System used in the Sierra Vista Elementary Schools. The funding for this CAI system was provided under ESEA Title I. It's appropriate, therefore, to spend a few minutes reviewing the Title I grant system.

The federal government has established a program of additional instruction for children who are "educationally disadvantaged". Examples of such children are those who have missed large amounts of school due to illness, those who have changed schools frequently and those whose native language is not English. Many children who have these problems are able to maintain their classroom work but those who fall behind are considered to be eligible for Title I assistance. This program is not a part of the Special Education program which provides extensive work with the child who is handicapped. Funds for Title I programs are determined by the number of low income families in the school district. This number is calculated from the results of several statewide surveys. However, eligibility for the program is not determined by income; it is based on the child's school achievement. Therefore, a child from the area's wealthiest family could be considered "educationally disadvantaged" while children from the state identified "low income families" are not necessarily eligible for Title I aid. The funds for programs in Arizona are administered by a Title I Divison Director who is located in Phoenix.

The 1975-76 Title I program in Sierra Vista consists of nine tutors to work with first, second and third grade students in reading and ten computer terminals to be used by students, grades 3-6, in reading and language and by students, grades 1-6, in mathematics. Three computer "monitors" were hired to staff the computer rooms in each of the elementary schools. Additional staff included a reading specialist and a math consultant who were to coordinate classroom and Title I efforts. Figure I illustrates Sierra Vista's organizational relationship of Title I and other elements of the district. A total of $132,000.00 was allotted for this school year's Title I activities. Approximately $40,000.00 of this money was used to lease the computer aided instruction system from Computer Curriculum Corporation. This is the first Title I CAI system in the state. Indications are that we will be funded to continue next year in order to better evaluate the results.

Sierra Vista is located in the southeastern part of Arizona, about 80 miles from Tucson. The population is 20,000 which includes 8,000 persons living on Ft. Huachuca. The major employer in the area is the federal government. A majority of the Sierra Vista working population is connected with Ft. Huachuca as active duty military, Civil Service or support personnel. There is also a growing community of retired military persons. The Sierra Vista School District has nearly 5,000 students. Children living on Ft. Huachuca attend county run accommodation schools until ninth grade at which time they enter the Sierra Vista system. The district itself has three elementary schools, one junior high school (grades 7-9) and one senior high school.

One problem facing the school district is a high student turnover. Department of Defense employees, both Civil Service and military, may stay for only six months. In addition there is a large amount of building in the area, many children of construction workers follow their parents from job to job. As in any rapidly developing community, school facilities are not quite keeping pace with the increasing school population. It was recognized that there were students in the district who, although not eligible for the Special Education program needed instruction in addition to that which they were receiving in the classroom. A decision was made to begin to remedy this situation at the elementary level first. The result was the Title I proposal for both a tutoring and a CAI system.

The computer aided instruction will now be discussed in more detail.

All the hardware (that is, the computer and its associated electromechanical equipment) was furnished by Computer Curriculum Corporation (CCC) under a package agreement. The computer itself is a model A-16 minicomputer, built by Cincinnati Milacron, the parent corporation of CCC. One disk drive provides mass memory. Student/machine interface is via ten ASR-33 teletype terminals. Four of these terminals are located with the computer at Carmichael Elementary School; three each are remotely located at Bella Vista and Village Meadows Elementary Schools. The remote terminals are linked to the computer by dedicated telephone lines leased from Mountain Bell. Figure 2 is a block diagram of the computer system. All required maintenance is performed by Computer Curriculum Corporation as part of its contract.

All system and application computer programs (that is, the software) plus curriculum materials are also provided by CCC. Details concerning program structure are considered proprietary by CCC and cannot be discussed here.

The elementary curriculum packages used by Sierra Vista Public Schools provide users with three services:

A. Student Information Entry and Updating
B. Student Progress Evaluation and Reporting
C. Student Drill and Practice

Each service will be described in detail.

A. Student Information Entry and Updating
After a child has been selected for the CAI program, he is assigned an identifying number. No other student in the system will have this number. The following information is then entered:

1. Child's first and last name
2. The primary classroom teacher's identifying number (A second teacher's number is optional; this feature allows more than one teacher to receive the child's weekly report.)
3. The CAI course in which he/she is to be enrolled (reading, language arts or mathematics)
4. The starting grade level in the course (The level is determined by scores from two achievement tests which the child took before he/she entered the Title I program. The beginning level is approximately six months lower than the test scores would indicate. This guarantees that the student will have a high initial success rate and will build the confidence often lacking in poor students.)
5. If the student is being enrolled in mathematics, the length of each session must be determined. This length may vary from five to ten minutes. All reading and language arts sessions are ten minutes long. The math course also has a variable timeout feature (the amount of time a student has to answer each question). Reading and language arts have a fixed timeout of 60 seconds, while the math course timeout may range from 30 to 60 seconds. We use the 30 second timeout for all students except children working on a first or second grade level. They are given a 60 second timeout.
6. If a student is being enrolled in more than one course, return to step 3 and continue.

An example of a student's enrollment is Figure 3.

If the entered information needs to be updated (for example, a child changes classrooms or adds a course), a similiar routine is followed. The old information is displayed; the new typed in to replace the outdated information.

All entry data must be done using a password. In our system, this password is available only to the math consultant and the computer monitors. No one else is able to enroll students.

The computer system itself updates a student's grade level. During the first ten sessions in a course, a student's grade level may change a half year per session. If he/she answers less than 50% correctly, the grade level is decreased by six months. If the score is more than 95% correct, the grade level is increased by six months. After the initial ten sessions, students advance and decline at a slower rate. Each course is divided into different strands; the child moves in each strand at his/her own pace. For example, the mathematics course has 14 strands. A child in the first grade might be working on five of them. The grade levels might be:

1.5	Numerical Concepts
1.2	Horizontal Addition
1.4	Horizontal Subtraction
1.4	Vertical Addition
2.0	Vertical Subtraction

The average score for a course is a weighted average of the strand scores. A child enrolled at the fourth grade level in mathematics would be working in 13 strands. In reading there are six different strands while in language arts there are seven strands. In a course the child progresses in each strand according to his/her performance in that particular subject area.

B. Progress Evaluation and Reporting

The classroom teachers receive as a weekly report a printout of their students' current grade levels. This report can be run at any time interval since the students' level is updated by the computer system following each session completed by that student. A day by day report, however, generally shows little change and simply increases the amount of paperwork handled by the teacher. The computer monitor also keeps a copy of this weekly report with her in the computer room to be available if the principal or other official would like a quick report of the students' progress. The math consultant and the reading specialist keep cumulative reports of all CAI students' weekly progress. These records will be used in evaluations of the Title I program this summer. Reports can be keyed by student, by class and course, by class or for the total Title I enrollment. Figure 4 is an example of a class report. The report routine does not require a password; in our practice, no one runs the report routine except the math consultant and the three computer monitors.

C. Student Drill and Practice

A student accesses the system by turning the teletype to on-line and pushing the space bar. The computer responds "Number and Name, Please". The child types in his/her identifying number and first name. If the computer recognizes the student as a valid user, the child's last name is printed along with the date, time and a "Hello" message. The child must then type in the initial of the course in which he/she is going to work. The entering routine was readily mastered by our only first grader using the system in less than one week; it is not difficult to handle. The computer then types a question for the student to answer. The reading and language arts lessons require either a one word answer or a number response. If the wrong answer is entered, the result is "//////The answer is . . . . .". A "timeout" (a delay of over 60 seconds) is considered an incorrect response. Figure 5 is an example of a reading lesson. The math lessons respond to a wrong answer with "Try again". If a second incorrect response is made, the system returns with "Type___#___" (the correct digit), so that the student eventually sees the correct answer. Responses to problems written horizontally are made left to right. Vertical problems require right to left entry for the answers. This allows students to answer the problems in the order that they are calculated. Scratch paper and pencils are available if they are needed. Figure 6 is an example of a math lesson.

At the end of the allotted time for the lesson, the number of correct answers is listed and the student is invited to continue. If the student indicates (by striking the space bar) that he/she wants no more lessons at that time, a "Goodbye" message is printed. The student takes the lesson printout to his/her classroom where, ideally, the classroom teacher goes over the problems with the student. Children often take the papers home to discuss them with his/her parents.

Students were selected for the CAI program mainly by teacher recommendations. The criteria used by the teachers were: 1) the child could not be on the official roster of Special Education students; and 2) the child must be performing below classroom level in reading, mathematics or language arts. Scores from the Iowa Test of Basic Skills (ITBS), administered in early September, were suggested as guidelines for selection. Originally, the teachers were told to refer only three students. Later, as the program was better established, more students were added. Currently we have 210 students in the CAI system. Thirty of them are enrolled in more than one course.

The terminals arrived at the elementary schools at least one week before they were installed. Therefore, many students were aware of them and were asking questions about the terminals. Children who were to be in the CAI program were introduced to the computer over a two day period and were allotted about twenty minutes for a brief general description of the system and their first "hands-on" experience. The reaction to this exposure was almost unanimously astonishment, especially when the computer "knew" their last name. The monitors sat with the students for the first three days to encourage them and remind them of what action was required next. After that, the children were ready to work on their own. The monitors still remain in the room but students are encouraged to do all their own work. The monitors are instructed not to give answers to the children. However, they often will review the questions with a child after the lesson is complete. A large number of students have experienced difficulty with the measurement problems (money, weights, lengths, etc.). The monitors have made special aids to help in this area. There are either posters with various measuring relations or handouts with similar information on them. In most instances there is a warm relationship between the monitor and their students.

After five months the initial bubbling enthusiasm for the computers is still evident in at least ten percent of the children. While the great majority of students enjoy their CAI lessons, it has become a routine feature of their regular school day. One sixth grader could not tolerate her computer lessons; the pressure she felt left her near tears after every session. Her teacher talked to the girl and found that she wished to discontinue her CAI time. We were agreeable and she was removed from the enrollment. After several months some students appeared to be bored. In order to counteract this we did several things. As an incentive toward higher scores, a plan was set up to recognize the students who missed no problems on a day's lesson. At Carmichael School, there is a chart in the computer room listing each child in the CAI program. A red square is placed after a student's name each time he/she received 100% on the lesson just completed. At Bella Vista and Village Meadows Schools, "Happy Grams" are given when a child has a perfect lesson. These "Happy Grams" are telegram sized papers with smiling faces printed along the edges. A message is written by the monitor congratulating the child on his/her achievement. "Happy Grams" are also given to those students who, although they didn't make 100%, did much better than their usual score. Another method that is effective in rekindling a dying interest in CAI, is to allow the child to work on fixed strands or topics. Fixed strands is an option of the mathematics course which allows the child to work on only one strand per session. Most of the sixth graders spent two or three weeks on the fractions strand. Fourth graders may benefit from and enjoy spending a few weeks on multiplication drill. Topics is a feature similar to fixed strands but it works in the area of language arts. A child can be enrolled in contractions or noun-verb agreement for example. It seems that a student's interest increases with a week or two of concentration in one specific area. However, the interest continues at the higher level even after they are returned to their regular program. We are working into other means of motivating and continuing to motivate the CAI students.

Although only the students who are below classroom level are working in the program, there's been no problem with children being embarrassed or feeling inadequate because of this special attention. In fact, the opposite seems to be occurring. The children are proud that they are chosen to work on the terminals and are often the envy of their classmates because of their opportunity to do so.

Most parents learned of the CAI program through their child. There has also been a local television interview on the system and articles in the Sierra Vista and Tucson newspapers. The CAI program has been discussed at PTA meetings. At one Carmichael School PTA meeting, the parents were allowed to complete a lesson themselves. At Bella Vista and Village Meadows Schools, letters were sent to the parents of the CAI students inviting them to visit the school at their child's scheduled computer time and see the program in action. We had a good response (better than 50% at Village Meadows) with many parents taking time off from work in order to see this new method of doing lessons. The children enjoyed having a chance to teach their mothers and fathers how to use the terminal. Most of the parents said that they were pleased that their child had an opportunity to work with the computer. There is a standing invitation to parents to visit the computer room at anytime during the school day.

Although the program is still young, we already have several success stories, for example:

> Troy H., a second grader, was unable to master even the simplest addition or subtraction facts. He was placed in the CAI program. After six weeks, he had not only caught up but was ahead of most of his class. Since he had accomplished his goals, he received a "Happy Gram" from his monitor stating he had completed his computer course and was "graduated" back to his regular classroom.

> Annalisa B., a third grader, scored at a grade level of 2.4 in the arithmetic section of the Wide Range Achievement Test (WRAT). That was ten months below grade level. She spent 43 sessions in the math program and increased her arithemetic WRAT score to 3.6 which put her one month above grade level.

Our most notable successes have been our Vietnamese students. Most were in the area for only a few months and were unable to be pre or post tested because they understood English so poorly.

> Quang Ho, a sixth grader, who worked in the program from November 25 to January 30 increased his course grade level from 3.5 to 5.0.

The official teacher orientation was through a one day seminar conducted by CCC. Our Title I program was funded and operational by late September, more than a month after school had started. It would be an impossible task to get substitutes for all of Sierra Vista's elementary teachers. Therefore, a decision was made to have the seminar presented to the grade chairmen from each of the elementary schools, a total of 18 teachers. The presentation consisted of a slide presentation of CCC's results in other school districts, an explanation of the Title I program as it was to be run in Sierra Vista, and some "hands-on" experience for the group. The idea behind choosing the grade chairman to view this program was that they would present the material to the other teachers in their grade. Some did an excellent job, some did nothing and most were somewhere in between. Therefore, the classroom teachers received a wide range of information or misinformation. This, I believe, explains in part, the variety of responses by the teacher to the CAI system. I feel that the teacher's use of the program is the single most important factor in affecting the child's progress. If the lessons are used as they should be, as a teaching aid, the student can make great strides. By reviewing each day's lesson, it may be seen that the child has a specific problem, such as being unable to borrow. Then that particular area can be reinforced by the teacher. Our CAI system is not intended to teach concepts; it is to be used for drill and review. Alone the CAI lessons are of more benefit than a page of arithmetic problems or a language assignment. There is immediate feedback after a child's response. The difficulty of the lessons is adjusted as the student progresses. This can be used by the teacher to discover the child's gaps in knowledge.

Most of the teachers collect the printouts as the child returns to class and go over them as time allows. One second grade classroom incorporates the work of its three CAI students into its regular mathematics lesson.

The weekly reports are used mainly as general indication of the students progress. Many teachers have said that the reports indicated their students were working on a lower level than they realized. This is often caused by gaps in a child's knowledge which can be hidden when working in the classroom. If scores in one or two strands are noticeably lower than the others, perhaps extra drill in those areas is indicated.

Academic progress made by the CAI students can be measured by three different means: 1) the grade level as computed by the CCC system; 2) pre and post Iowa Test of Basic Skills (ITBS) scores and 3) pre and post Wide Range Achievement Test scores (WRAT). Usually the ITBS is given in early fall, however, the Sierra Vista Schools are changing to spring testing. Therefore, this school year we will have two ITBS results. The Iowa tests are given schoolwide; there is no makeup date. The WRAT test is given as a child enters and leaves the Title I program. The only exceptions are if a child speaks no English or if he/she moves without giving us notice. The grade level calculated by the CAI system as based on the students performance is available every week.

The establishment of the Title I program for the elementary schools has had an additional advantage. The math consultant and reading specialist are available to the classroom teachers who have problems or questions in which they would like an outside opinion. There is now a group responsible and able to work with those students that seem to fit nowhere, they are not handicapped and therefore could not be helped by the Special Education program, but they are not performing adequately in the classroom.

Now that the Sierra Vista Schools are on their way to becoming veterans in CAI usage, we feel we have conquered some problems and are better able to cope with others. There was, for nearly six weeks, almost continuous difficulty with the modems (the electronic devices connecting terminals and the computer to phone lines). The two remote location schools, Bella Vista and Village Meadows, suffered a lot of downtime. This was discouraging to both monitors and students. The children were reluctant to come to the computer room because so often when they did one or more of the terminals were inoperable and they would have to return to class without working on their lesson. Fortunately, the problem was solved by acquiring a new type of modem. The referral system will go more smoothly now that lines of communication have been established. Next year we hope to be able to fill at least half of the computer slots by the second week of school. Attendance has been a problem for a few students. It is important that the child do a computer lesson each school day, if at all possible. Of course, during special times such as the week before Christmas vacation and Valentine's Day, there may be schedule conflicts. But as a general rule a child in the CAI program should complete a lesson each day he/she is in school. The best motivating factor seems to be teacher support. If the classroom teacher thinks that it is important that the child "go to the computer" each day, chances are that the child will go. However, when the teacher really doesn't care, the child tends to "forget" two or three times a week. We feel that the attendance problem will be resolved when a better system of informing teachers of CAI capabilities and applications are developed. There is no workshop week previous to the opening day of the school year in Sierra Vista. The teachers are in school only one day before the students return. Therefore the CAI orientation will need to be during the first week of school. A meeting in the computer room with the teachers of each grade while their classes are at music or physical education (30 to 45 minutes) should be sufficient for a brief information session plus some "hands-on" time. The importance of teacher follow-up will be stressed. They will then be in a better position to decide if they wish to use the CAI system as a teaching aid. It should be made clear that our CAI is in no way going to supplant the teacher or be responsible for higher student/teacher ratios. It is a method of enabling a student to have additional drill and practice in the specific areas in which he/she needs help.

A school system establishing a computer aided instruction system must consider a number of questions and make some rather difficult decisions. As always, costs will be an important component. First and foremost, it must be established that the CAI programs will meet the school's needs and that the CAI philosophy is compatible with the school's traditional materials. Here, for once, cost should not be the ruling factor. An expensive program that meets your needs is probably worth the extra money. Next, it must be determined that the associated documentation and support materials are adequate. Finally, flexibility and usefulness of system generated reports should be considered. The school system should insist that the software vendor provide continued consultation and expanded documentation as required. Schools which anticipate unusual requirements should consider contacting for custom modifications to standard software. I strongly recommend that only extremely prosperous and well qualified school systems attempt internal development of computer aided instruction programs. Development of all types of software is notoriously difficult and time consuming and can be staggeringly expensive.

Once a curriculum has been selected, attention must be given to hardware. The quickest and simplest solution is to contract for a package, as Sierra Vista Public Schools did with CCC. Our vendor provided a complete system, hardware, software, maintenance support, and staff training. Standard packages, however, are never quite what the user would like. An experienced school system might do well to tailor the hardware somewhat. The first choice is whether to use an existing computer or to acquire a new machine to be dedicated to CAI. This is a very difficult question, depending on a host of matters such as how easily the software can be adapted to the existing computer. The decision should be made by experts after careful study. If the experts aren't available or time doesn't allow a thorough study, follow the software vendor's recommendation.

The other important hardware consideration is terminals. CAI terminals is between a keyboard printer combination (like a teletype) or a keyboard cathode ray tube (CRT) combination. The CRT units have the advantages of being silent, not requiring paper, and being essentially nonmechanical. Furthermore, they look something like television sets and may be more attractive to TV conditioned children. A considerable disadvantage is that no permanent record ("hard-copy") of the student's work is produced. Keyboard printer units, of course, produce hard-copy. Standard ASR-33 type units are very cheap and quite reliable. They are somewhat noisy and require messy chores like changing paper and ink ribbons. Some of the more expensive units, using thermal printers, are fairly quiet and don't need ribbons. While good cases can be made for almost any type of terminal, plain 33-type units are probably best to start with. Whatever type of terminal is selected, the terminals should be located in a special room, apart from ordinary classrooms. This relieves the classroom teacher from the added responsibility for the machine and makes for more efficient scheduling of terminal usage.

Finally, maintenance must be provided for all hardware. Again, a package agreement is simplest. In smaller centers like Sierra Vista, there may be no other way of obtaining qualified maintenance personnel. However maintenance is provided, insist that rapid response be a condition of the contract.

The size and type of staff required to administer a CAI program will, of course, vary from district to district. The amount of paperwork is large--remember that Title I is a federally funded grant program. Therefore, we need a full time secretary. A CAI program funded by a different method might not. A monitor is necessary at each terminal site. Someone needs to handle changing paper and ribbons, reminding students of what to do next and fielding the little unexpected problems that always occur. In addition, our monitors give the WRATs as children enter and leave the CAI system. There must be a person who is in charge of the CAI program. If this should be his/her sole job would depend on the size of the CAI system. Our Title I Director is also a school psychologist and an administrator in charge of school nurses, counselors and special education teachers. Therefore, the math consultant serves as the acting CAI Director although that is not her title. Curriculum specialists would again depend on the size and extent of your CAI program. An advantage of such specialists is that there is a central person to turn to with special problems. If such a person is already available in your district, perhaps it would not be necessary to duplicate the position for the CAI program. In Sierra Vista we have a Title I psychologist who gives psychological evaluations to many of our CAI students. We have found this to be a very helpful and informative service. Adequate lines of communication between CAI and existing school district staff are most important. Person-to-person communication with additional memoranda on important points seem most effective. Principal and teacher orientation is most important and should be carefully planned. All teachers and administrators should be included in this area. Student selection will be determined by the purpose of your program. Our guidelines were set up by Title I. Perhaps another CAI system would be geared toward the gifted child or made available for drill for all students. Here it is necessary to have your objectives well in mind before student selection begins. Parents should be strongly encouraged to become acquainted with the program. The best method to reach as many persons as possible is to have both daytime and evening opportunities to view the terminals. Data derived from the CAI system may provide information which could be used to strengthen the existing curriculum in the subject, school or district. There will be a wealth of material available; it should not be ignored.

In summary, Sierra Vista Public Schools, with federal financial help, has found that computer aided instruction is a valuable educational tool. A straightforward program with modest goals has provided nearly 300 students with drill and encouragement not available in traditional environments. As a bonus, the CAI support staff has been able to provide consultation and analysis services previously lacking. Successful CAI, we have found, calls for careful planning and program selection, followed by extensive orientation of teachers, parents and participation of students.

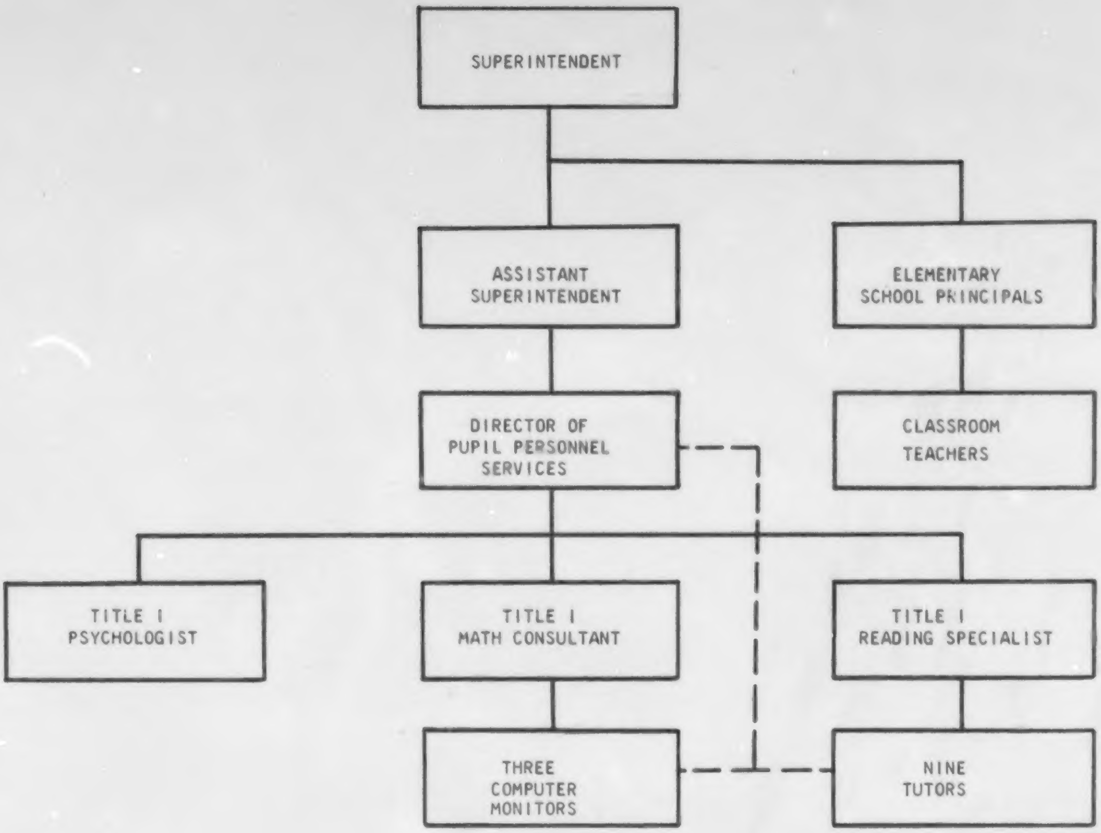

SUPERINTENDENT
ASSISTANT SUPERINTENDENT
ELEMENTARY SCHOOL PRINCIPALS
DIRECTOR OF PUPIL PERSONNEL SERVICES
CLASSROOM TEACHERS
TITLE I PSYCHOLOGIST
TITLE I MATH CONSULTANT
TITLE I READING SPECIALIST
THREE COMPUTER MONITORS
NINE TUTORS

FIGURE 1 The organiztion chart of Title I and other school personnel, Sierra Vista, Arizona.

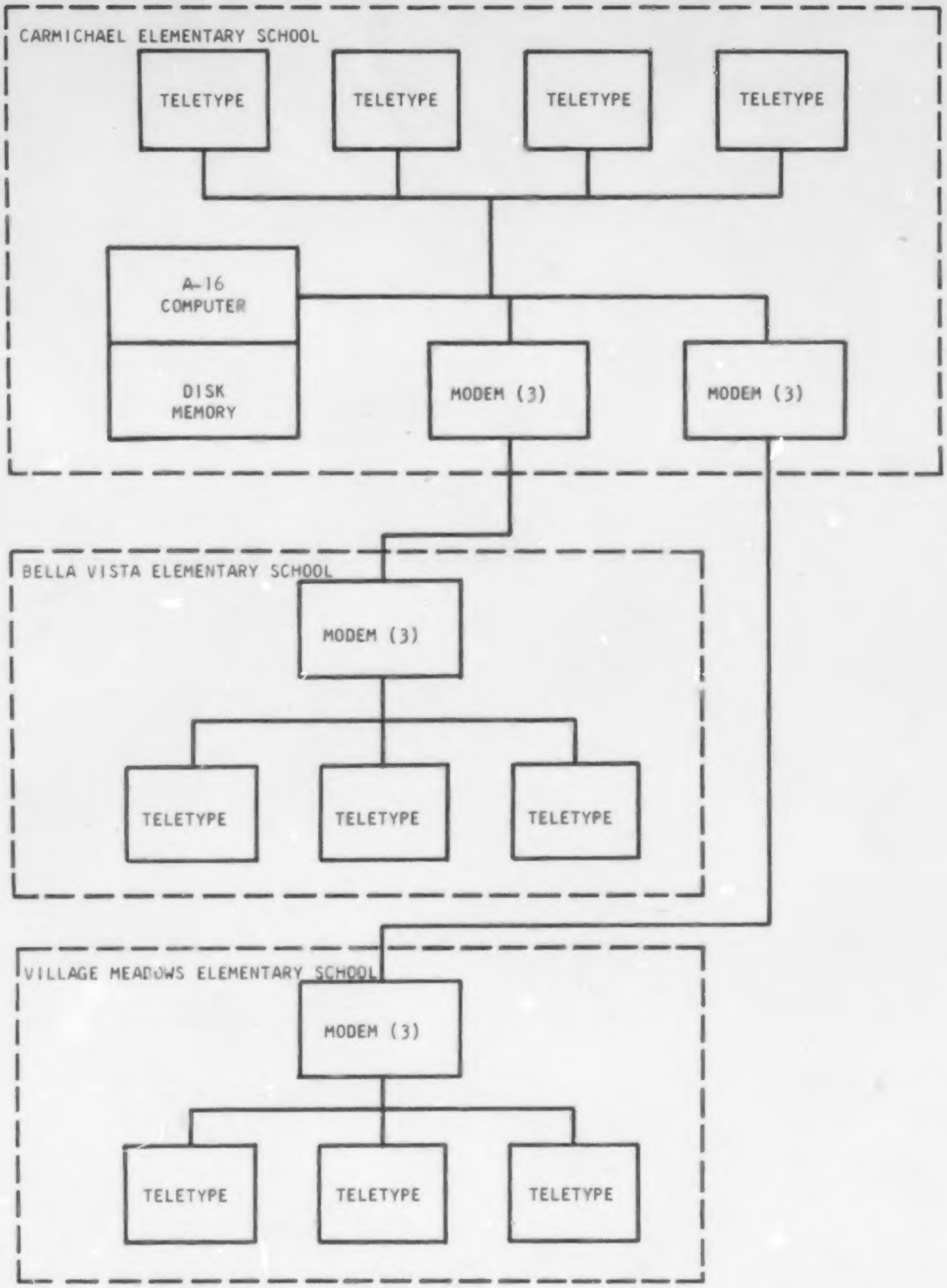

CARMICHAEL ELEMENTARY SCHOOL
TELETYPE
TELETYPE
TELETYPE
TELETYPE
A-16 COMPUTER
DISK MEMORY
MODEM (3)
MODEM (3)
BELLA VISTA ELEMENTARY SCHOOL
MODEM (3)
TELETYPE
TELETYPE
TELETYPE
VILLAGE MEADOWS ELEMENTARY SCHOOL
MODEM (3)
TELETYPE
TELETYPE
TELETYPE

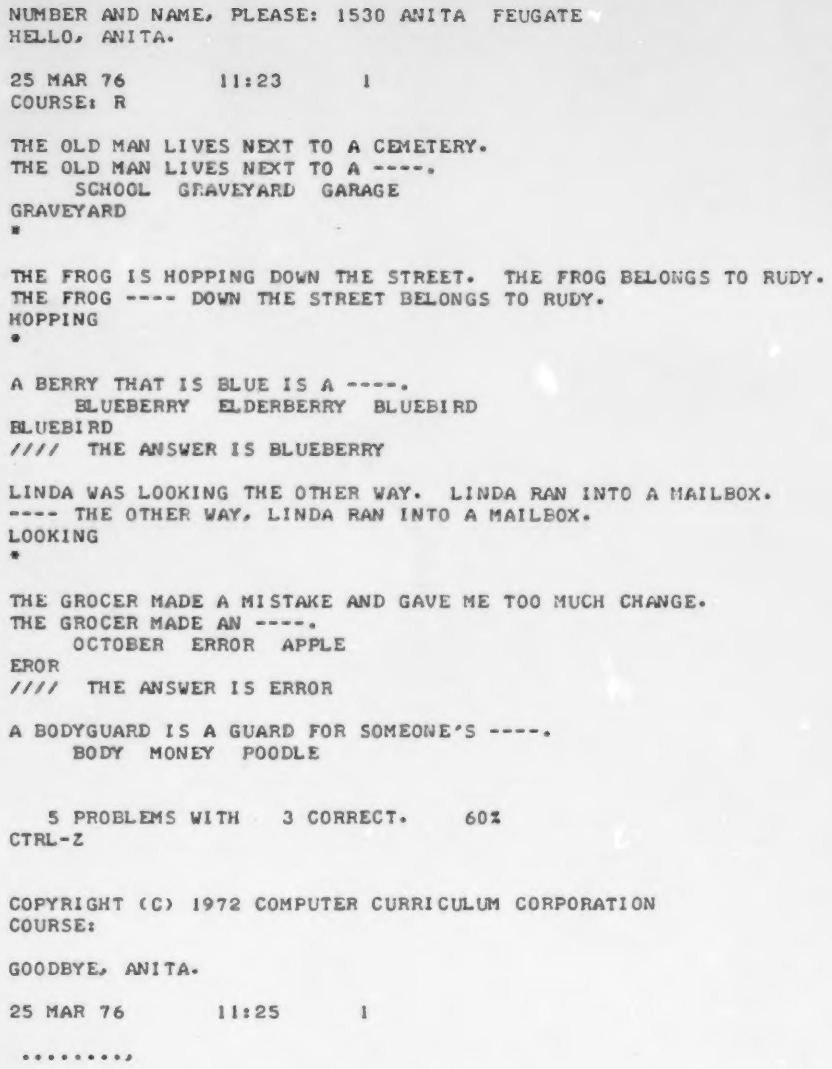
FIGURE 2. The Computer Aided Instruction network in use at Sierra Vista.

```
NUMBER AND NAME, PLEASE:  P[[[[[
WHICH ROUTINE? STUDENT
STUDENT NUMBER: 1222
NAME: NORMAN CHESTNUT
CLASS: 2-3
COURSE: R
GRADE LEVEL : 3.5 - 40
COURSE: M
GRADE LEVEL:     10   38
SESSION:      7
TIMEOUT:     30
COURSE:
STUDENT NUMBER:

25 MAR 76        11:12       1

 .........,
```

Figure 3. An example of a student enrollment.

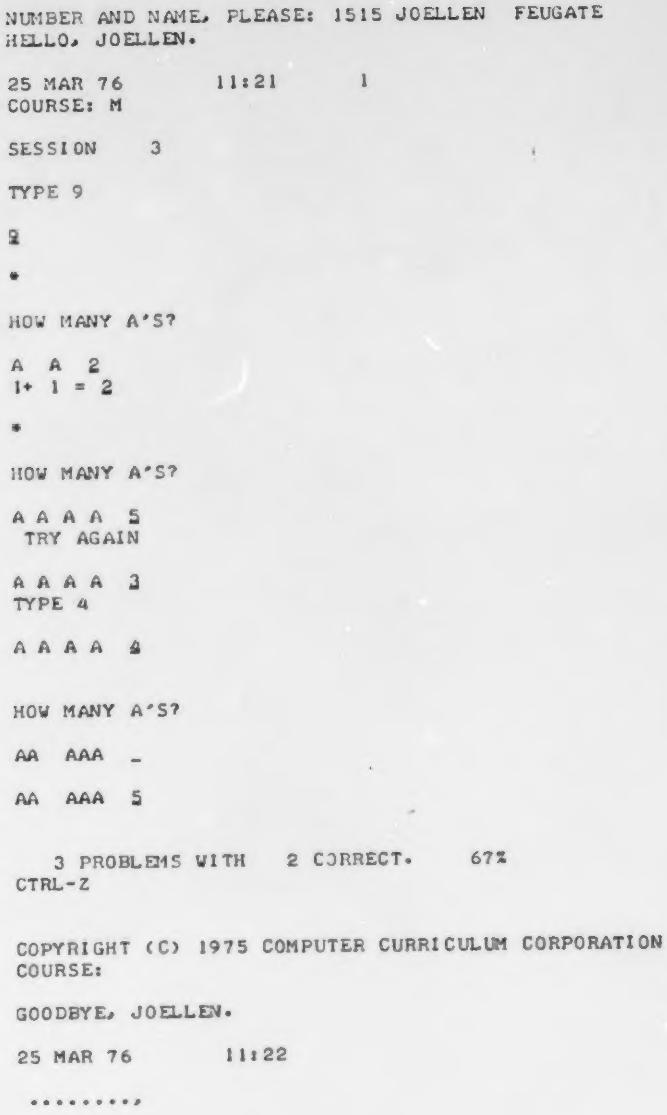
```
NUMBER AND NAME, PLEASE:  REPORT
REPORT TYPE (S/C/T): C
REPORT NUMBER: 12
COURSE: X




        12    SYWASSINK   5

25 MAR 76       11:14      1
MATH
ID#   FIXED    SES   TIME AVG  NC HA HS VA VS EQ MS HM LW VM DV FR DC NG
  27+  VM36     76   1:11  20  20 20 20 20 20 20 20 -- -- -- -- -- -- --
    DEBBIE BABUCA
  28+  VM36     75   1:18  20  20 21 20 20 20 20 20 -- -- -- -- -- -- --
    DON KANAWAH

25 MAR 76       11:15      1
READING
  HR:MN  TO  BS  WA VOC  LC  IC  WS AVG    ID
   1:26   1     4.0 4.0 4.0 4.0 4.0 4.0     2    ALEXANDRA LENTZ
   1:43   0 2.6                     2.6     3    LARRY TOWNSEND
   1:38   1     3.5 3.5 3.5     3.5 3.5     5    DANNY CHANDLER

25 MAR 76       11:15      1
LANGUAGE STRANDS
NUMBER   A   B   C   D   E   F   G AVG   HR:MN

25 MAR 76       11:15      1
TOPICS
  ID  TCL TOPIC 1   2   3   4   5   6   7   8  AVG

REPORT TYPE (S/C/T):

25 MAR 76       11:16      1

 ........,
```

Figure 4. An example of a Weekly Class Report.



```
NUMBER AND NAME, PLEASE: 1530 ANITA  FEUGATE
HELLO, ANITA.

25 MAR 76       11:23      1
COURSE: R

THE OLD MAN LIVES NEXT TO A CEMETERY.
THE OLD MAN LIVES NEXT TO A ----.
     SCHOOL  GRAVEYARD  GARAGE
GRAVEYARD
*

THE FROG IS HOPPING DOWN THE STREET.  THE FROG BELONGS TO RUDY.
THE FROG ---- DOWN THE STREET BELONGS TO RUDY.
HOPPING
*

A BERRY THAT IS BLUE IS A ----.
     BLUEBERRY  ELDERBERRY  BLUEBIRD
BLUEBIRD
////  THE ANSWER IS BLUEBERRY

LINDA WAS LOOKING THE OTHER WAY.  LINDA RAN INTO A MAILBOX.
---- THE OTHER WAY, LINDA RAN INTO A MAILBOX.
LOOKING
*

THE GROCER MADE A MISTAKE AND GAVE ME TOO MUCH CHANGE.
THE GROCER MADE AN ----.
     OCTOBER  ERROR  APPLE
EROR
////  THE ANSWER IS ERROR

A BODYGUARD IS A GUARD FOR SOMEONE'S ----.
     BODY  MONEY  POODLE


   5 PROBLEMS WITH   3 CORRECT.    60%
CTRL-Z


COPYRIGHT (C) 1972 COMPUTER CURRICULUM CORPORATION
COURSE:

GOODBYE, ANITA.

25 MAR 76       11:25      1

 .........,
```

Figure 5.   A sixth grade level reading lesson.



```
NUMBER AND NAME, PLEASE: 1515 JOELLEN  FEUGATE
HELLO, JOELLEN.

25 MAR 76        11:21       1
COURSE: M

SESSION    3

TYPE 9

9

*

HOW MANY A'S?

A  A  2
1+ 1 = 2

*

HOW MANY A'S?

A A A A  5
 TRY AGAIN

A A A A  3
TYPE 4

A A A A  4


HOW MANY A'S?

AA  AAA  _

AA  AAA  5


   3 PROBLEMS WITH   2 CORRECT.    67%
CTRL-Z


COPYRIGHT (C) 1975 COMPUTER CURRICULUM CORPORATION
COURSE:

GOODBYE, JOELLEN.

25 MAR 76        11:22

 .........,
```

Figure 6.   A first grade level mathematics lesson.

AUTHOR LANGUAGE SIMPLICITY: AN ANSWER TO CURRICULUM DEVELOPMENT

Paul Furnas
Sperry Univac Educational Data Systems, St. Paul, MN 55165

ABSTRACT: Curriculum development represents a substantial step in the operational and economic costs of a CAI system. Sperry Univac directly addressed the problem by creating a logical and simple English-based author language. The Author language for Education and Training (ASET) uses several innovative techniques to make it both easy-to-use for the inexperienced CAI author and flexible and powerful for the experienced author. The language transforms teachers and instructors into CAI authors from existing faculty without specialized pre-training in computers or programming.

Introduction

From the moment of its birth in the 1950's, Computer Aided Instruction (CAI) has promised great potential for progress in this key social activity. Somehow, realization of its potential fell short of the vision cast by the sum of its parts. One of the persistent roadblocks to full fruition has been an operational and economic one: curriculum development.

Sperry Univac directly addressed this problem in their new Author System for Education and Training (ASET). As the name implies, ASET revolves around the central ingredient in any educational system, the curriculum author. ASET begins with an author language that transforms teachers and instructors of existing faculty into authors without specialized training in computers or programming. Sperry Univac achieved this breakthrough by building into the language a logical and simple, English-based syntax. For example, the language contains a two level structure: one for beginners and a second command repertoire with a full, powerful CAI structure for experienced authors that is an extension of the first level. The beginner's level enables a faculty member to render his or her specialty into a complete Socratic dialog lesson suitable for presentation to a student with just 3 commands.

A few words further explain the ASET project group's concept of simplicity. Simplicity was not chosen as a key design objective arbitrarily but because of its impact on the human factors engineering of the externals of the author language. The exigencies of training users in a higher level language soon teaches the basic premise that simplicity is the outstanding factor that one can select among several human factor features. Others include congruency with existing language systems familiar to the expected population of users, appropriateness in terminology to the educational profession, minimization of the explicit signals required from the user to implement the functions from the system that the user wants, and maximizing the number of functions the system selects automatically by default to provide the option used by the highest frequency user, capable of override in the unusual case.

The following sections trace in greater detail the varied ways the concept of simplicity threads throughout the ASET System. These include:

1) Hierarchial structure in two levels for beginner and experienced user. Only 3 commnads initiate the beginner.

2) English based, classroom oriented syntax.

3) Fill-in-the-blanks organization for lesson development.

4) Lesson structure analogous to writing textbooks.

5) Conversational dialog option for lesson development.

6) Automated functions with author overrides.

7) Transparency of computer imposed constraints such as memory allocation, word size, etc.

8) User oriented documentation.

Two Level Hierarchial Structure

The difficult aspect of learning any language is the problem of getting started. Once started, each level learned becomes a springboard to the next and more advanced level. ASET solved the initial inertia problem by designing the complete system with the basic simplicity that an author could begin with a knowledge of no more than three commands. With these three commands, an author's usual writing skills, and a simple pre-printed form; the author renders his or her specialty into a complete Socratic dialog that will appear on the student's terminal. The three commands are:

IF

PRINT

GOTO

These three selected commands are the backbone of all the ASET commands and are powerful enough in themselves to guarantee the author a simple but adequate lesson on his or her first attempt. The three commands accomplish one basic CAI cycle: presentation of concept or information to be learned, statement of question to test student's perception of concept, evaluation of student's response, and the supply of reinforcement to the student (negative if incorrect and positive if successful), and direction to the next logical step in the learning sequence depending on the results of the evaluation of the student's response.

The IF command instructs the computer to evaluate the student's response to the author's testing of the student's perception. In ASET the author need not command the computer's display of the author's concept, the related question in the Socratic style, nor the reception of the student's answer. These operations are all performed automatically by the computer without author command. The author's IF statement sets up the criteria for acceptance or rejection of the student's answer.

The PRINT command allows the author to reinforce the response by the student. The author provides two PRINT commands for each CAI cycle at least, one if the answer is the answer anticipated and one for an incorrect response. In ASET, the author may even have these two commands automated for him or her and allow the computer to reinforce the student's answer without any explicit instructions.

Finally, the GO TO command allows the author to individualize his or her instruction by directing the flow of the lesson to one of any number of alternative next steps in the lesson according to how the author sees the instructional needs of this particular student. In other words, if the student needs some practice before the student attempts the question again, the GO TO commands request the necessary sequence from the computer. If the author judges that a certain response requires remedial or supplementary instruction, a GO TO command directs the lesson flow accordingly. The GO TO commands determine the path that a student takes as he or she proceeds through a lesson. Each student proceeds through the lesson in a unique manner. When executed by the computer, the GO TO command directs the student to the most efficient and effective learning for that student.

The GO TO command also provides two paths from each question. If the student answers the question correctly, the author provides a GO TO command that branches the student to the next unit within the lesson sequence. The correct response path or 'prime' path, provides the most direct route through a lesson. If a student answers incorrectly, the author provides a second GO TO that branches the student backwards to repeat part of the lesson or to execute a remedial sequence.

In summary, the GO TO commands enable the author to control the flow of the lesson. Occassional uses branch a student to practice sessions, pre-test exams, and final examination.

English Based, Classroom Oriented Syntax

Any higher level language must satisfy the needs of the computer for preciseness and direction. The temptation or tendency is that the language becomes more oriented inward to the realities of such computer operations or characteristics as word size, registers, memory allocation, memory addressing, and memory transfer. ASET avoids these tendencies. ASET looks outward to the author user and couches its terminology in terms of a language system familiar to the classroom teacher.

An alphabetical list of commands demonstrates both the simplicity of the language and its basic English language system orientation:

ANSWER
AUTO
BACK
CLEAR
DELETE
DO
END SESSION
ERASE
EXIT
END HELP
HELP
PROJECT
RESPONSE TIME
RETRY
RETURN
SET
SKIP
TAB
UPPER
RANDOM

The options selected by an author show a similar character:

COMMENT ON
ANSWER ON
BACK ON
RESPONSE TIME
UPPER ON
AUTO NO ON
AUTO YES ON
RETRY
SESSION TIME
ITEM ANALYSIS
PHONETIC ON
FORMAT
MICROFICHE
AUTHOR
ARRAYS
ANTICIPATED RESPONSES

The parameter strings associated with the commands and options show a corresponding attention to simplicity. Most are limited to no more than one parameter. The parameters consist of OFF or ON, an integer number, or a simple label.

Fill-In-The-Blanks Structure

Each Socratic dialog cycle in the ASET system consists of a framework of three elements. The three elements appear explicity on a line by themselves in each lesson unit; thus, serve as prompts to the author. When the author has filled in the three elements, all of the required elements of concept presentation, follow-up question, response evaluation, reinforcement, and guidance to the next logical step needed for one completed Maieutic cycle are present.

The three elements are:

Initial text

COMMANDS

TEXT (reinforcement message)

The initial text may be from 1 to 12 lines in free form of the author's choice. The COMMANDS element consists of strings of commands the author selects from the list of available commands to control lesson flow, to handle student responses, to delete portions of the student response, to manipulate variables, to give or withhold student exercise of optional activity, to restart, to select amounts of records to be kept, and to evaluate responses by the student. The TEXT element contains all of the reinforcing replies the author anticipates, both negative and positive.

The fill-in-the-blanks character of the basic lesson unit structure, besides having the virtue of assisting beginning authors, also ensures that experienced authors consistently take advantage of the potentials of sound CAI principles of instruction.

Analogies to Writing Textbooks

In the final analysis, the writing of an ASET lesson looks to the prospective author as a task that is very similar to writing a conventional text book. In a faculty that contemplates extending their educational offerings to CAI or enriching their existing curriculum with CAI, the likely candidates for authors are those who have already written textbooks or are likely to.

A paragraph of a conventional textbook looks very much like the initial text of an ASET lesson. The question a textbook author would locate at the end of a chapter or in a section on exercises at the end of the textbook, is located immediately following the presentation of the concept or fact in the ASET lesson.

A set of ASET units combine to form an ASET lesson analogous to a chapter in a conventional textbook. A set of lessons go together to make an ASET course completing the analogy to the conventional course textbook.

The textbook author develops the technique of identifying and communicating a subject in his or her area of specialization in terms of readily explainable discrete steps. This same skill is readily transferable to segmentation of the author's subject into the basic units of an ASET lesson, a cycle of Maieutic dialog. The combination of several units fits naturally into both an author's feel for the flow of his subject and the flow of the computer's execution of an ASET lesson.

Development of Lessons Through Conversational Dialog

The persuasive practicality of [illegible] Socratic method has the effect of alerting those who work with it of the frequent opportunities to use it in contexts other than the conventional teacher-student dialog. Such an opportunity became very obvious early in the ASET project effort. The idea was to use the method to assist new authors in the use of ASET. The result was a conversational mode that an author could use to create an ASET lesson.

The Conversational Mode proceeds by displaying a series of questions on the author's terminal. The author types in the answers to the questions on the terminal keyboard and the system does the rest. The questions are simple and brief so that they can also serve as prompts for further lessons after the author's first. Besides the feature of teaching a prospective author the use of ASET, the Conversational Mode has the advantage of methodically ensuring that the potentials of the CAI approach to learning are built into the author's lesson.

To illustrate the simplicity and brevity as a constructive adjunct to system design, the following are the Conversational Mode prompts. They also give in a larger sense the author's progressive development of an ASET lesson.

```
LESSON OPTIONS?
# RETRIES ALLOWED:
RESPONSE TIME (SEC)
SESSION TIME (MIN)
ALLOW CALC?
ALLOW COMMENT?
ALLOW ANSWER?
**UNIT LABEL:
UNIT OPTIONS?
# RETRIES ALLOWED:
RESPONSE TIME (SEC)
TEXT:
QUESTION:
ITEM ANALYSIS?
ANTICIPATED RESPONSE:
ANALYSIS TYPE:
TOLERANCE ALLOWED:
REPLY:
BRANCH:
REPLY TO UNEXPECTED ANSWER:
BRANCH:
EDITING?
DISPLAY SYNTAX?
OK?
BUILD UNIT?
ASSEMBLE LESSON?
LISTING?
SAME FILE, ELEMENT NAMES?
FILE, ELEMENT NAME:
```

Readers may see that the following prompts from the above set represent one cycle of a Socratic dialog:

```
TEXT:
QUESTION:
ANTICIPATED RESPONSE:
ANALYSIS TYPE:
TOLERANCE ALLOWED:
REPLY:
BRANCH:
REPLY TO UNEXPECTED ANSWER:
BRANCH:
```

The prompts related to a cycle repeat reiteratively until the author concludes a lesson. The last eight prompts allow the author to review the lesson, make any changes desired, and to specify the documentation for the author's record. Finally, the system automatically supplies all the commands that the author's answers to the prompts have implied. As can be seen, to develop a lesson using the Conversational Mode requires no specialized computer knowledge at all, not even the three basic ASET commands noted earlier as the minimum needed. The author only needs the usual skills of written communication and the knowledge of the author's area of specialization.

Automated Functions with Author Overrides

As a deliberate objective of design policy, the ASET System minimizes the number of explicit signals required from the user to implement the functions the user wants. To give this amount of automation to the author and still give the author control, the automatic functions are generally accompanied by an override for the exceptional case. The high frequency options execute automatically; the less frequent option requires the explicit command, directive, or parameter. This approach assists the beginner author particularly. As the author gains experience with the system, he or she may call on more and more of the customizing power of selecting options that meet the specialized needs of the author's own concept.

A few examples give the thrust of how this design goal effected the ease of use of the ASET System:

1) Initial text and author's follow-up question do not require commands for their display to the student.

2) The reception of the student's answer, its processing, and the control of the frame advance are all performed automatically without specification by the author. All computer buffers, memory allocation, memory addressing, register setting, word sizes, etc., are all transparent to the user in the input-output required by the sequence of computer events.

3) The system automatically initiates the next lesson unit in the lesson sequence unless the author specifies otherwise.

4) If the author does not specify a parameter, the system sets an average value in default. For example, the system sets the limit for a student to respond to a question at 90 seconds, an average value. If this is not satisfactory to the author, he or she may specify any value from 10 seconds to 999 seconds. The system automatically sends a message to the student when the time limit expires.

5) The author may specify functions for the lesson as a whole rather than reiterate them for each unit. For exceptions the author may specify an override command on a unit-by-unit basis. For example, the author may specify the number of allowable retry attempts to answer a question. The author specifies this once in a Header Directive rather than as many as 512 times if this is the number of units in the lesson.

6) An author may initiate a full record keeping and report generation system with only two steps: selecting the type and amounts of records to be kept on-line while the student takes the lesson and the content and extent of analyzation of the data from these records for the final report.

7) The system automatically erases the information left over from the previous lesson unit before displaying the initial text for a new lesson unit. The author may override this feature if the circumstances require it.

8) The system automatically processes most student key-ins on the terminal keyboard of an administrative nature. The author is relieved of providing allowance for this type of eventuality.

9) Some services that a computer can uniquely provide are available that would take a great deal of ingenuity on the author's part and hours of time. An example is a random number generator. Another example is a phonetic match generator.

Transparency of Computer Imposed Restraints

The ASET author's language is congruent, not with computer science language systems, but with an English based language system. A good example of this orientation is the absence of any computer imposed limitations in the expressions that the author uses. The inner workings of the system accomodate all of these restrictions such that the author is completely unaware of them; in other words they are transparent to the author.

Examples include:

1) Word size
2) Byte size
3) Input-output buffers
4) Memory allocation
5) Memory addressing
6) Memory medium
   (core, disk, tape)
7) Register assignment
8) Register saves
9) Character mapping

All of the above are set or manipulated on a common sense basis of expected average values, adjusted by operating experience, with provision for overflow when needed.

User Oriented Documentation

Much software documentation is written from the viewpoint located at the center of the system looking outward. For example, the principle subdivisions of the software system become the principle subsections of the user's guide. The ASET user's documentation is written from the viewpoint of the user looking in from the outside of the system. For example, the principle subsections of the user's guide are the questions in the user's mind as they occur to the user in a logical order to accomplish the CAI function. The subsections become a step-by-step guide for writing a lesson.

Another example of user orientation of the documentation is the continual visual correlation maintained between what the author writes in the fill-in-the-blanks development of the lesson and what the student sees as a result. Complete examples exist with graphics that in a sequence of frames show a complete lesson in a cinematic style.

The ASET documentation gives repeated overviews or summaries of the author language. In effect, an author can take short courses or review the language in two or three pages. Frequent tables organize the relevant information in such a way that an author may detach them from the parent document and keep them for handy reference. A pocket guide, of shirt or vest pocket dimensions, serves as a reminder of the details of the system for the experienced author that he or she can take to the computer console or terminal for convenient, quick look-ups when functioning on-line with the system.

Conclusions

The simplicity of the ASET Author Language is deceptive. Behind a simple-to-use, classroom-oriented, external design there beats a very powerful tool for the utilization of recent discoveries into the science of the process of learning. The 20th Century witnessed a rigorous application of the searchlight of the experimental method to learning theory. Built into the ASET system called into play by this terse, direct language, lies many of the fruits of this innovative research. Examples are:

1) ASET automates the positive reinforcement requirement postulated by B. F. Skinner as a key ingredient in learning. Reinforcement is individualized immediately and order synchronized with each student's attempt to learn.

2) ASET realizes, in practice, the time constraints for association through the simultaneity of stimulus and response as measured by Pavlov and Galton. The quicker the reply, the stronger the association.

3) ASET established rewards that are contingent on the behavior desired as proposed by Bloom.

4) ASET improves learning rates by incorporating active student participation as correlated by Witasek and Gates.

5) ASET gives an author the opportunity to repeat student practice based on correct responses rather than incorrect ones, a strategy suggested by Ebbinghouse.

6) ASET ensures a pre-planned, uninterrupted implementation of Bloom's, Krathwohl's, Simpson's, and/or Gagne's taxonomic behavioral objectives at each step in the instruction.

7) ASET includes a large proportion of psychomotor taxonomies as suggested by Simpson by the student muscular responses required to operate the terminal keyboard.

8) ASET multiplies the number of different individual skills, aptitudes, and experience of students that can be anticipated with a separate response to each. ASET accomodates all on an individual basis even though any one student takes only one branch.

9) ASET's flexibility provides new opportunities to contiguously present a concept, all operant rules about that concept, concrete examples, and examination as a strategy to enhance learning through discovery techniques as proposed by Gagne.

10) ASET lessons incorporate both the linear program method of Skinner and the variable program of Crowder. The author is free to select one or both for the learning method appropriate to the learning objective.

11) ASET increases student motivation through the deliberate arrangement of learning materials such that the frequencies of positive reinforcement are maximized and the frequencies of negative reinforcement are minimized.

12) ASET avoids the inefficiencies of overlearning and minimizes forgetting stimuli as defined by Ebbinghouse.

13) In ASET, learning progresses by developing the performance skill of overlapping operations in increasing order of complexity so that the total requires less time as suggested by Cattell.

14) ASET emphasizes the internalization of inner growth as the ultimate objective of education, the operant learning as Skinner theorized.

The educational innovations made by ASET, and the skill of the author, combine to provide the potential for students to learn faster and achieve higher test scores than they might achieve by conventional instruction. ASET offers new thresholds for building quality into educational programs, quality in terms of the enhanced skills of the students trained.

Bibliography

Computer-Assisted Instruction: A Survey of Literature, Third Edition, Albert E. Hickey, Entelek Inc., Newbury, Massachusetts, 1968, Chapter 5 Programming Languages

The Conditions of Learning, R. M. Gagne, Holt, Rinehart, and Winston, 1970

The Technology of Teaching, B. F. Skinner, Appleton-Century-Crofts, New York, 1968

Taxonomy of Educational Objectives, Benjamin S. Bloom, D. McKay, New York, 1956

DISCOURSE ANALYSIS FOR CAI

Sally Yeates Sedelow
National Science Foundation
and
University of Kansas (on leave)

ABSTRACT:

What is Discourse Analysis?
Who does it?
How is it done?
What will Discourse Analysis do for CAI?

In answering the above questions, this presentation will provide an introduction to discourse analytic techniques for natural language. Computer scientists, linguists, and literary scholars are all very much involved in research which has both short-term and long-term implications for the teaching of English and other natural languages through computer-aided instruction in educational instruction. Highlights of current research and its relevance for CAI will be stressed.

THE TYRANNY OF THE TERMINAL

by

E.O.G. Davies
Algonquin College, Ottawa, Canada

ABSTRACT: The time has come when the quality of courseware must be given serious thought. Hardware experts lead the thinking, and teachers tend to write for the hardware that exists rather than for the best educational methodology. Some proposals are made for more control on quality, and a few possibilities of first-language training are demonstrated.

It has become apparent to me, over the years, that computerized education has become, unintentionally perhaps, a by-product of the computerization of education.

Computers were used for educational administrative purposes before the possibility of their being applied to educational processes was realised. The hardware, then, existed, and the demand that came to be heard was "Where's the courseware?" Teachers began to comply.

But many teachers found they were in opposition to, rather than in co-operation with, the hardware experts. I was challenged: "What I want to know is, what can your programs do for my computers?"

The enormity of such illogicality from intelligent people is so mind-boggling that I am afraid that on that occasion I lost the opportunity to make a suitable riposte.

An institution "for studies in education" is on record as saying that it discouraged the planning of software that could not be used on existing hardware. It is not difficult to see the problems of an infant many regarded (and many still do regard) as a bastard, if it could not show that it had a right to existence. But one can see also the long-term implications for educational software.

A leader in research into the educational computer field told me that he had no philosophical concern for education: that was up to the academics. His particular approach to learning was so far ahead of much academic thinking that one can only shed a tear for the loss to education.

Further, he held so unswervingly to his tenets that hundreds of teachers were given the opportunity to computerize bad teaching. One recognises that such an approach avoids the confrontations that would be so tiresome and so dangerous.

But surely the time has come to challenge what is being allowed on to the computer. We are tired of seeing the same brilliant programs by the same brilliant teachers. It is time to look at all the others, and to recognise that there has to be some kind of control over the quality of programs, to acknowledge, despite all the theories, that some methods, some approaches, work better than others, and that quite simply, some do not work. Computerized education has grown up but has not yet matured. It is time, in the interests of our students - all students - to challenge, to confront. It is time to challenge the teachers - to challenge them to use the computer as the best of all learning tools, and not as just another visual aid - and it is time to challenge the hardware experts to produce hardware that will enable the student to learn in an environment that he cannot obtain in any other way; the terminal should create the educational environment. It should no longer be acceptable that software succumb to existing hardware, resulting so often in blinkered thinking, and eventually in another deadening process for the learner.

The need is for courseware that teaches as no teacher can, and for hardware that can enable courseware to teach as no teacher can.

There have been, I suggest, too many unwanted births, and there may have been too many abortions.

If the possibilities are greater than any teacher, however brilliant, the dangers are also greater than the danger posed by any teacher, however dangerous. One of the most dangerous of teachers, if not necessarily the most dangerous, is the one who, wittingly or unwittingly, attempts to produce thinking and acting copies of himself.

Am I proposing the setting up of some educational dictator? Resoundingly no; not even if it were to be myself. Perhaps especially if it were to be myself. Education, it seems to me, has an opportunity to become the servant of the learner in a way that, to the best of my knowledge, it never has been. The master has, in more senses than one, been the schoolmaster. But I am not railing at the grossly overworked and usually grossly underpaid teacher (overworked and underpaid if he attempts to handle his job with any seriousness and professionalism). He has been attempting to do an impossible task. For the first time - and here is the answer to that glib line "There is nothing new in education" - it is possible to help every individual student; it is possible not only to tend his needs, but to help him overcome his problems. We can be free at last of that psychological trap - "the class".

But how are we to respond?

One of the most exciting educational experiments I have taken part in was team teaching. Team teaching has everything - almost. It can cross the boundaries of academic subjects, and show how related many disciplines are (surely a process long overdue); it makes use of a variety of minds and expertise, and not necessarily of one person per discipline (unless you organise it that way); it can and should use a multimedia approach; and with several people combining in a team, it has the chance of some professional critical input, which the teacher in the classroom rarely has an opportunity of receiving, even if he wants it. We can have all that, and the individual student catered for.

It is a team approach that should enable us to become aware of our own problems as teachers, to alert us perhaps to a better approach, to question our occasional intellectual flabbiness, even at times to question our grasp of our own material or even subject. Above all, it enables us as teachers to continue to learn.

But such team-teaching involves enormous effort and vast amounts of preparation time. Television has made even out of the youngest pupils very sophisticated critics (at least to the point of expecting the slickness of a good production). It involves also the timetabling necessary to prepare as a team and to teach as a team. As a result, after one such experiment it tends to remain an experiment - however successful it was.

I am convinced that the cross-discipline team approach is just round the corner for computerized education. Once the work is done it is done, and it is always available (which is not to say it can't be updated). But such an approach will make great demands on hardware.

Meanwhile, I am working with a team that is not multidisciplinary, but is attempting to cope with learning in that most important of disciplines - communications. A student's success in all his studies depends on the development of his communications skills; as does his own personal development.

The demands being made on people are showing up their "weaknesses" in communications skills that were not apparent before because they were not needed. What was functional literacy and oracy is no longer so.

As teachers, we know that our aim must be to place the student in a position where he is less and less reliant on us, and more and more reliant on his own ability. But the methods used to achieve functional oracy and literacy, at any level, are important. Methodology has, in some quarters, become a dirty word; but it is vital that we use research findings and our experience in learning/teaching to produce courseware that does what we claim we are trying to do.

Research indicates that most adults assume that they are able to read, but in fact they fail to understand what the author is saying to them: their own opinions constantly get in the way, and the author is credited with views for which there is no supporting evidence. Somewhat surprisingly, this applies also to well-educated adults.

We are building a course of computerized English language training that includes all the skills of communication - listening, speaking, reading, writing, and thinking.

We see them as interdependent skills, and the student is helped to understand and to perform. He learns through the process how to learn; and I stress that it is a process that he is undergoing.

We believe that a student's growth must be rooted in his experience, and we attempt to involve the student both intellectually and emotionally, and to move him to intuitions and conceptualising, to make imaginative projections, to read not only between the lines but beyond the lines.

One of the most important principles of teaching, and the least possible under normal classroom conditions, immediate feedback, is given in such a way that the student is not only made fully aware of what he says but is helped to arrive at an acceptable answer. "Right" answers, incidentally, are not what is most important. What is most important is the process the student is engaged in.

The task of building programs of this kind forces us continually to define and redefine what good teaching is, and what the aims of education should be. Obviously, we are led to question our own teaching - a painful but necessary process.

I would like to discuss briefly the approach. We see language training as being not a handling of problems of remedial work, but essentially as developmental. We isolate an area in this process, and discuss how it is best handled i.e., how it can be best taught - how is it best learnt. If it is decided to use written material we search for something suitable.

The questions we pose on the material (incidentally, as will be seen, questioning on the material is only a part of the process) are validated by a class at the level for which the material is chosen. Two or three students are taken out of the class and they answer, individually, the questions orally - a dialogue ensues between them and the teacher, which is tape-recorded. The class as a whole meets the team and discusses the passage, the questions and the answers, and that also is tape-recorded.

With this experience, the team decides to drop some questions, introduce others, and alter the wording in some (where it can be seen the wording - not the concept - is at fault).

The validation process is gone through again with a different class at the same level. And so on until we are sure the material is doing the job we want it to, and our questions are handling the aspects of communication we want them to, and are also handling the problems the students have revealed.

The written answers are carefully sifted into Acceptable Answers, Partially Acceptable Answers, and Not Acceptable Answers. The answers in the last two categories are then examined for patterns.

We are ready to start writing our algorithms for the computer. The patterns of answers do, of course, give us our branching (we add extras to cover logical possibilities and to cover any other not-acceptable answer), and the tape-recorded sessions are a considerable help here, especially the one-to-one teaching which gives us the necessary break-down for individual teaching.

Some of the students' problems surprised us - as, for instance, when we discovered that they did not understand "yet" to be a qualification of a previous statement rather than a continuation. When we followed this up we were told by high school teachers that a majority had real difficulty in differentiating between "and" and "but".

One process at the terminal is as follows: the student is given the passage and is asked to read it out loud on tape as often as he likes, until he is satisfied that that is the best he can do. He can, of course, listen to his own recording(s). In our experience, students work very hard at this. When he is ready to proceed, he so indicates and we play him a recording made by ourselves. We stress that this is a reading, not the reading. He listens to our recording, and then re-listens to his own final taping. He can re-record and listen to any part of the whole as he wishes.

The questions are put to him when he is ready, and he engages in a dialogue on questions and answers. He is not only expected to have a response, but to articulate it.

Finally he takes part in a seminar. It is unusual for someone not to say something like "The master tape read such-and-such a part like this ... I still think it should be read like this ..." At once one is into the whole process of meaning - context, vocabulary, stress, rhythmic patterns, grammar, syntax, punctuation.

We find a student with a mind's ear, as opposed to a mind's eye, a rarity. And the ear is a vital part of language training.

We are engaged in research to use, along with a CRT or teletype, a random-access audio disc, a cassette tape-recorder, a voice-intonation print-out, a touch panel, Super-8 movie film, 35 mm. slides, and an interactive graphic terminal (3-D capability).

But at all times our concern is with what is the best possible way of teaching that particular stage; nothing is used because it is there. And if it isn't there and we think we need it, we take steps to probe the possibilities.

We find that students are motivated to learn through interest. If he finds his understanding is blocked by a problem, he is prepared to tackle his problem, even if it is one of grammar. It is, of course, only when we fail to understand that we need to examine the symbols and the structures.

Perhaps I can illustrate parts of the process with a passage from "Or I'll Dress You in Mourning",[1] a biography of El Cordobes, the Spanish torero. The passage is an exercise in implication. Apart from the factual information given, the reader is constantly being asked to anticipate - something an alert reader, and any listener, does by habit. It is, of course, good habits that we should be trying to help the student acquire.

> Las Ventas, as it had been for moments, was a roaring cavern of noise. Yet, in that wildly howling mob, a sense of concern began to rise in a few seasoned minds.
>
> Trying to accomplish the impossible feat of licking the paper of one of his hand-rolled cigarettes without taking his eyes from the man in the center of the ring, Don Juan Espinosa Carmona, the chaplain of the bullring, had the same thought. For several moments his anxious mind had turned over a nonstop invocation to the Virgin for the impetuous youth who had this afternoon stirred his feelings as no matador had done for two decades.
>
> Thirty ranks below him, unnoticed in the yelling crowd, a man slipped from a stool set alongside the rim of the callejon. While the crowd went crazy, Dr. Maximo de la Torre hobbled anxiously toward his infirmary.
>
> Theirs, however, were only a few warning voices in a sea of exultation. All Las Ventas, it seemed, was on its feet yelling, shouting. So was half of Spain, pulled into the bullring by a television cable. Cafés, bars and restaurants all over the country were in a pandemonium. Everywhere - in clubs, schools, stores, homes - people pushed and shoved, fighting for a glimpse of the screen, demanding, "What's he doing? What's he doing?"
>
> Those instants were the fulfillment of everything El Cordobes had ever wanted, the solace for every beating he had ever taken, the balm for every hunger pain he had ever known. He was, in the center of that bullring, the very quintessence of a "somebody." Twenty million people were calling his name, and yet he - he - was alone in a world of his own with a bull spinning around him.

[After passage had been read, tape-recording played]

The first question we ask is "What is Las Ventas?" and the second is "How do you know?" This is part of a dialogue that ensued between the computer and a student.

Computer: What is Las Ventas?

Student: Las Ventas is the bullring or arena

Computer: That's right, Wayne.
How do you know? Quote the first piece of evidence that tells you without doubt what Las Ventas is, and then say what paragraph it appears in.
The order is important and both parts must be in the answer.

Student: A roaring cavern of noise paragraph one

Computer: Once you have read the passage you know what the "roaring cavern" is. But at this point in your reading you do not know, although you may suspect. This is part of the art of reading. You pick up clues and hold them in your mind until you are certain.
What makes you certain and in what paragraph is it?

Student: The centre of the ring, paragraph 2

Computer: "Ring" is not definite enough - it could be any kind of ring. Try again.

Student: The chaplain of the bullring, paragraph 2

---

[1] By Larry Collins and Dominique Lapierre; Simon and Schuster, New York, 1968.

```
Computer: O.K., that's fine.
          In Funk & Wagnall's 'Standard College Dictionary' the word
          "cavern" is given as:
               Cav·ern (kav'ərn) n. A cave, especially one that
               is large or extensive.  -v.t. 1. To shut in or as
               in a cavern.  2. To hollow out.  (MF caverne,
               L caverna, cavus - hollow)
          If you do not understand the meaning of n., v.t., MF, or L
          press the "H" key for help.  Otherwise press the "Return" key.

Student:  [Return pressed]

Computer: Why do the authors describe Las Ventas as a "roaring cavern
          of noise"?
          Is it because:
               A) They want you to see Las Ventas in your mind's eye?
                  OR
               B) They want you to hear the sound in your mind's ear?
          Type in either the letter A or B.

Student:  B

Computer: Yes, good.
          They are attempting to create the impression of noise.  "Noise"
          is a key word, reinforced by "roaring".  "Cavern" does, of
          course, give us the same visual sense but in agreeing that the
          image is essentially auditory, can you tell me how the word
          "cavern" can help us to understand better the 'noise' that was
          produced?

Student:  Noises made in a cavern will echo and rebound off the walls

Computer: Very good.
          The sound would be increased by its reverberation, as the
          authors are saying is happening in the bullring.
          O.K.  Now tell me what "v.t." (in the dictionary definition)
          stands for.

Student:  I don't know

Computer: Oh!  I thought you knew!
```

Figure I is one of two algorithms that enabled the dialogue to take place.

A student's problem - even if it is that of guessing rather than working from understanding, as he should be doing - is handled. He is able to apply immediately what he has learned, and he is helped to become aware of the reading process. If he has no problems he is not held up.

The next question is concerned that the student is aware of the importance of sound, of his <u>ear</u> in reading, the auditory imagination. The question is deliberately drawing attention to the dictionary definition, to familiarize students with the way dictionaries present definitions, to encourage the use of dictionaries, and to show that one still has to use intelligently the meaning of the word in the context. (Figure II).

We ask why the use of "mob" rather than "crowd", what "seasoned" means, why Carmona attempts an impossible feat, the words that indicate excitement, and a phrase that suggests something different; and all the questions lead eventually to "From your understanding of the passage, do you think El Cordobes will kill the bull, or do you think he will be gored by the bull?" The answer is not just guesswork; all the clues are there, and the student has been made aware of them.

The attention is focused on importance of sound in understanding by such passages as the one we take from "A Christmas Carol." It is not just a matter of the <u>meanings</u> of words:

[The tape-recording is played first, then the passage is given]

> Oh! But he was a tight-fisted hand at the grindstone. Scrooge! a squeezing, wrenching, grasping, scraping, clutching, covetous, old sinner! Hard and sharp as a flint, from which no steel had ever struck out generous fire; secret, and self-contained and solitary as an oyster. The cold within him froze his old features, nipped his pointed nose, shrivelled his cheek, stiffened his gait; made his eyes red, his thin lips blue; and spoke out shrewdly in his grating voice. A frosty

> rime was on his wiry chin. He carried his own low temperature always about with him; he iced his office in the dogdays; and didn't thaw it one degree at Christmas.
>
> [The tape-recording is played again after reading.]

We reverse our earlier procedure not just to make a change: quite simply, we have found that that is the best approach here, and it is vital not to fall into patterns and stereotypes. We are concerned with rhythms of the written word: students expect rhythms in poetry but seem not to expect them, and certainly fail to respond to them, in prose.

Falling and rising rhythms are reasonably basic; but we meet the inevitable problems by using music. A spin-off here is that some students, who have never listened to anything other than "rock", find that there do exist other kinds. Oh yes, we'll use rock as well. All education is manipulation - but we are aware of some of the dangers, as I have pointed out. We found that after the passage on El Cordobes, many students wanted to know more about bullfighting - not unnaturally - but they were also asking questions on Spain, its people, its music, its architecture. We used discs, tapes, slides, to satisfy their curiosity (and to further stimulate, of course). It suggests exciting possibilities for the team approach to computer education: all should be there, the hardware should be able to handle it, and the student should be able to pursue an interest by pushing a button. Many people have had life-long interests triggered by similar accidents, and have had opportunities for following them up. How many have had an interest stimulated and no opportunity to pursue it? How many more have failed to be stimulated? (Figure III)

Figure IV shows an algorithm that handles sound somewhat differently.

Here are a couple of other passages we use:

> (a) I give a sideways glance at the mirror, and see a puffed face purpled with veins as though someone had scribbled over the skin with an indelible pencil. The skin itself is the silverish white of the creatures one fancies must live under the sea where the sun never reaches. Below the eyes the shadows bloom as though two soft black petals had been stuck there. The hair which should by rights be black is yellowed white, like damask stored too long in a damp basement.
> Well, Hagar Shipley, you are a sight for sore eyes, all right.[2]
>
> (b) Upon the ecstatic diving board the diver,
> poised for parabolas, lets go
> lets go his manshape to become a bird,
> Is bird, and topsy-turvy
> the pool floats overhead, and the white tiles snow
> their crazy hexagons. Is dolphin. Then
> is plant with lilies bursting from his heels.[3]

The questions we ask on the first are:

> (1) Quote any one sentence from the first paragraph and tell me what you find good about it. (We accept any answer).
>
> (2) What does one usually mean when one says that "something is a sight for sore eyes"?
> Do you think that is what is meant here? Answer Yes or No.
> Give me a word that is given to statements of that kind.

I am not prepared to run these passages to death. The student should begin to understand that good reading is for enjoyment, not for asking questions about, by pesky schoolteachers or cruddy computers.

We offer a student who has enjoyed a piece of writing the opportunity to hear some more. We don't force it on him. And we give him just so much, in the hope that, if a novel, he will get the book and read it, or, if poetry, he will look up some more poetry by the same author, or even just read some poetry.

Remember also I stressed that we considered all the communications skills as interdependent. We look forward to the student writing - it is surprising (although I suppose it really should not be) what they are capable of producing under the right stimulus.

---

[2] The Stone Angel by Margaret Laurence:
McClelland and Stewart Limited, Toronto; 1968.

[3] Lines from The Lone Bather, The Rocking Chair and Other Poems by A.M. Klein;
Ryerson Press; 1948.

We do, of course, have some idea of a framework for all this - narrative, descriptive, and so to expository prose. A fair number of students, using their textbooks, fail to remember what the first part of the sentence said when they reach the last part. Many cannot link sentence to sentence. By the time our students reach expository writing, they have already had many of their problems handled - the story line being a valuable connecting link.

Logically, it may seem sensible to start with the parts and build to the whole. But in practice the parts do not motivate the student (only the teacher knows the relationships), they tend to become exercises (and therefore "exercisy") and, bogging down on the mechanics, students rarely reach the goodies. Our students work hard to understand the niceties involved in descriptive writing because of the awareness created by descriptive touches in narrative. At this level of expository writing - the giving of information - students have nothing but their interest in the subject and their reading ability to carry them along.

With the passage below, we do not ask the student to listen: if the training has meant anything he should be able to read and understand.

With narrative, the training of the ear to catch correct phrasing, the modulation of the author's words, are important. The author helps the reader by adding words that give the reader information about the way the character spoke, what the speaker intended, the effect what he said had on the person he was speaking to, and so on. The writer of scientific prose does not use this complex signalling apparatus; he relies on an unambiguous order of words, the meaning of which is kept as simple as possible, and punctuation. But even here the reader almost always gains something in understanding if he reads it as if the author were saying something to him. It is not usually just a case of noting data and of following logic.

> In the leg there are two bones, the tibia and fibula. The tibia or shin-bone is long and strong and bears the weight of the body. The fibula or splint bone is an equally long but much slenderer bone, and is attached to the tibia as a pin is to a brooch. At the top, the tibia presents a broad smooth surface, which articulates with the lower end of the femur forming a hinge-joint at the knee. In front of the knee is a small separate bone, the patella or knee-cap. Here, again, the bones are fastened together by strong check ligaments which limit movement and prevent displacement. Both the tibia and fibula expand below into prominences of bone which can be felt on either side of the ankle-joint. Between these there is an articulating surface into which the top of a foot-bone called the astragalus fits, thus forming the ankle-joint. As the prominences are placed on either side of the joint they prevent the displacement of the astragalus. The ankle-joint is a hinge-joint, and is surrounded and secured by strong ligaments.[4]

Space does not allow me to go into this program in detail. But I would like to point out that the student is made aware that the writing of good scientific prose is an art (involving considerable skill) and can be (by some "has to be") learned. He understands not only how it is done the way it is, but why.

We do ask questions on grammar; but they are questions that lead to more important questions. And if he has forgotten his grammar, or is not too sure, we branch him for revision (Figures V and VI). The dialogue here now is critical. This, it seems to me, is the place for grammar. It is not taught in a vacuum, with all the alienation we are familiar with; researches have long ago shown that the study of grammar in isolation is a waste of time. But unfortunately too many teachers assume that the study of grammar is a waste of time.

Spelling? There are vastly more important skills in communication than spelling. But if a literate public can recognise illiteracy only as a problem of spelling, so be it. We have built a complete inductive basic spelling program; and of course it is true that, whereas spelling is best learned as one is learning to read, many students can be helped by a knowledge of a few basic spelling rules.

One example of our approach must suffice. We want the student to become aware that where the final syllable of a word is (a) stressed, and (b) has a single vowel before a single consonant (e.g. bat, regret), that consonant is doubled before a suffix beginning with a vowel (e.g. -ed, -ing). We first revise vowels, and short and long vowel sounds. We then teach compound words, root words, prefixes, suffixes, plurals, with branching where necessary. The algorithm (Figure VII) shows what the student does after this preliminary work. The exceptions are handled, and he is tested on his ability to apply what he has learned.

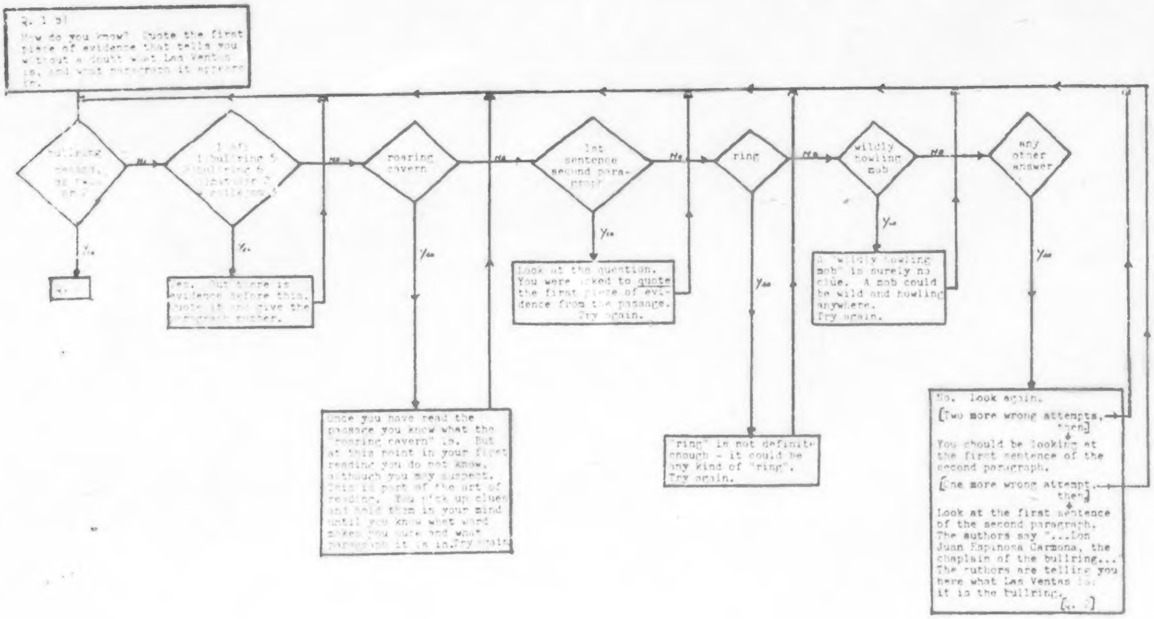
---

[4] Manual of Human Physiology by Leonard Hill;
Edward Arnold, London, England.

I have tried to illustrate some ways that we are using the computer to overcome classroom learning problems. We have in hand other programs that, with the cooperation of the National Research Council of Canada, are pushing the hardware to provide the, as we see it, necessary learning environment.

I have not been suggesting that we have all the answers. I have been posing some of the problems, and some solutions that we have found to work.

There has to be a developmental structure, but it must leave teachers free to use their initiative, to build their own programs, handling their students' problems. The teacher's function is bound to change, and the computer can release him to do things that are necessary and that he can do better than the computer. But it is to be hoped that he will accept the challenge not only of the computer but of his whole teaching methodology.



Q. 1 b)
How do you know? Quote the first piece of evidence that tells you without a doubt what Las Ventas is, and what paragraph it appears in.

Yes. But there is evidence before this. Quote it and give the paragraph number.

Once you have read the passage you know what the "roaring cavern" is. But at this point in your first reading you do not know, although you may suspect. This is part of the art of reading. You pick up clues and hold them in your mind until you know what word makes you sure and what paragraph it is in. Try again.

Look at the question. You were asked to quote the first piece of evidence from the passage. Try again.

"ring" is not definite enough - it could be any kind of "ring". Try again.

A "wildly howling mob" is surely no clue. A mob could be wild and howling anywhere. Try again.

No. Look again.

[Two more wrong attempts, then]

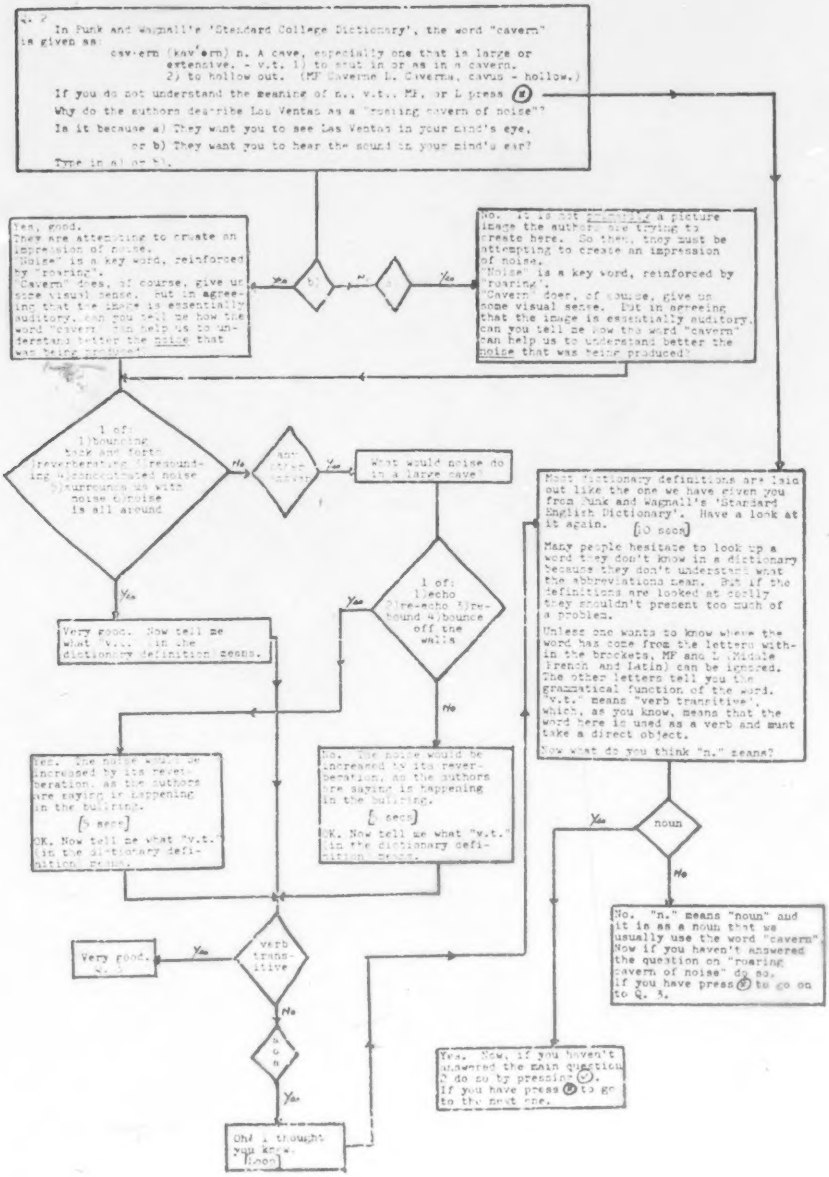
You should be looking at the first sentence of the second paragraph.

[One more wrong attempt, then]

Look at the first sentence of the second paragraph. The authors say "...Don Juan Espinosa Carmona, the chaplain of the bullring..." The authors are telling you here what Las Ventas is: it is the bullring.

[Q. 2]

Figure I


Yes
No
b)
a)
any other answer
1 of: 1)bouncing back and forth 2)reverberating 3)resounding 4)concentrated noise 5)surrounds us with noise 6)noise is all around
1 of: 1)echo 2)re-echo 3)rebound 4)bounce off the walls
verb transitive
noun

Q. 2
In Funk and Wagnall's 'Standard College Dictionary', the word "cavern" is given as:

cav·ern (kav'ern) n. A cave, especially one that is large or extensive. - v.t. 1) to shut in or as in a cavern. 2) to hollow out. (MF Caverne L. Caverna, cavus - hollow.)

If you do not understand the meaning of n., v.t., MF, or L press ⓧ

Why do the authors describe Las Ventas as a "roaring cavern of noise"?

Is it because a) They want you to see Las Ventas in your mind's eye,
or b) They want you to hear the sound in your mind's ear?

Type in a) or b).

Yes, good.
They are attempting to create an impression of noise.
"Noise" is a key word, reinforced by "roaring".
"Cavern" does, of course, give us some visual sense. But in agreeing that the image is essentially auditory, can you tell me how the word "cavern" can help us to understand better the noise that was being produced?

No. It is not primarily a picture image the authors are trying to create here. So then, they must be attempting to create an impression of noise.
"Noise" is a key word, reinforced by "roaring".
"Cavern" does, of course, give us some visual sense. But in agreeing that the image is essentially auditory, can you tell me how the word "cavern" can help us to understand better the noise that was being produced?

What would noise do in a large cave?

Very good. Now tell me what "v.t." (in the dictionary definition) means.

Yes. The noise would be increased by its reverberation, as the authors are saying is happening in the bullring.
[5 secs]
OK. Now tell me what "v.t." (in the dictionary definition) means.

No. The noise would be increased by its reverberation, as the authors are saying is happening in the bullring.
[5 secs]
OK. Now tell me what "v.t." (in the dictionary definition) means.

Very good.
Q. 3

Oh! I thought you knew.
[loop]

Most dictionary definitions are laid out like the one we have given you from Funk and Wagnall's 'Standard English Dictionary'. Have a look at it again.
[10 secs]

Many people hesitate to look up a word they don't know in a dictionary because they don't understand what the abbreviations mean. But if the definitions are looked at coolly they shouldn't present too much of a problem.

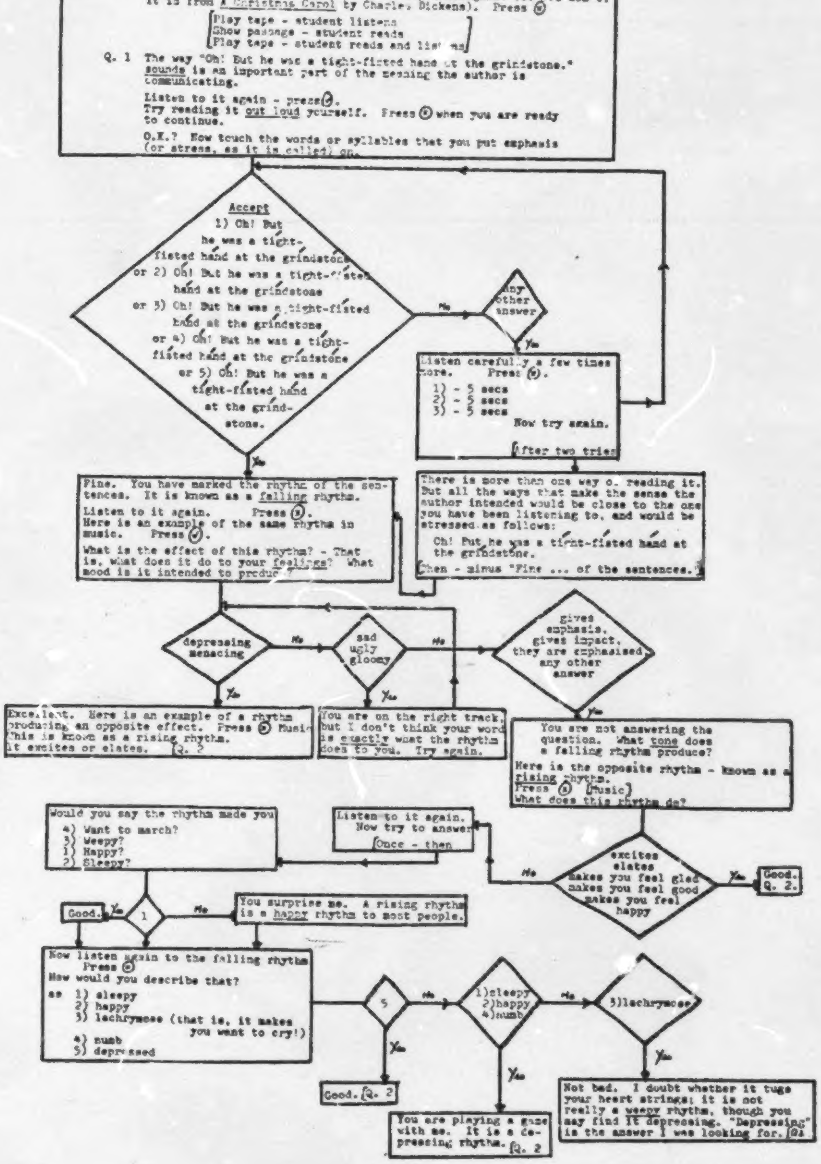
Unless one wants to know where the word has come from the letters within the brackets, MF and L (Middle French and Latin) can be ignored. The other letters tell you the grammatical function of the word. "v.t." means "verb transitive", which, as you know, means that the word here is used as a verb and must take a direct object.

Now what do you think "n." means?

No. "n." means "noun" and it is as a noun that we usually use the word "cavern". Now if you haven't answered the question on "roaring cavern of noise" do so.
If you have press ⓧ to go on to Q. 3.

Yes. Now, if you haven't answered the main question 2 do so by pressing ⓥ.
If you have press ⓧ to go to the next one.

Figure II



Passage 1. Listen to this passage, which I am sure you recognise (if you don't, it is from A Christmas Carol by Charles Dickens). Press ⓧ

[Play tape - student listens
Show passage - student reads
Play tape - student reads and listens]

Q. 1 The way "Oh! But he was a tight-fisted hand at the grindstone." sounds is an important part of the meaning the author is communicating.

Listen to it again - press ⓧ.
Try reading it out loud yourself. Press ⓧ when you are ready to continue.

O.K.? Now touch the words or syllables that you put emphasis (or stress, as it is called) on.

Listen carefully a few times more. Press ⓧ.
1) - 5 secs
2) - 5 secs
3) - 5 secs
Now try again.
[After two tries

There is more than one way of reading it. But all the ways that make the sense the author intended would be close to the one you have been listening to, and would be stressed as follows:
Oh! But he was a tight-fisted hand at the grindstone.
Then - minus "Fine ... of the sentences."]

Fine. You have marked the rhythm of the sentences. It is known as a falling rhythm.
Listen to it again. Press ⓧ.
Here is an example of the same rhythm in music. Press ✓.
What is the effect of this rhythm? - That is, what does it do to your feelings? What mood is it intended to produce?

Excellent. Here is an example of a rhythm producing an opposite effect. Press ⓧ Music
This is known as a rising rhythm.
It excites or elates. [Q. 2

You are on the right track, but I don't think your word is exactly what the rhythm does to you. Try again.

You are not answering the question. What tone does a falling rhythm produce?
Here is the opposite rhythm - known as a rising rhythm.
Press ⓧ [Music]
What does this rhythm do?

Would you say the rhythm made you
4) Want to march?
3) Weepy?
1) Happy?
2) Sleepy?

Listen to it again.
Now try to answer
[Once - then

Good.

You surprise me. A rising rhythm is a happy rhythm to most people.

Good.
Q. 2.

Now listen again to the falling rhythm
Press ⓧ
How would you describe that?
as 1) sleepy
2) happy
3) lachrymose (that is, it makes you want to cry!)
4) numb
5) depressed

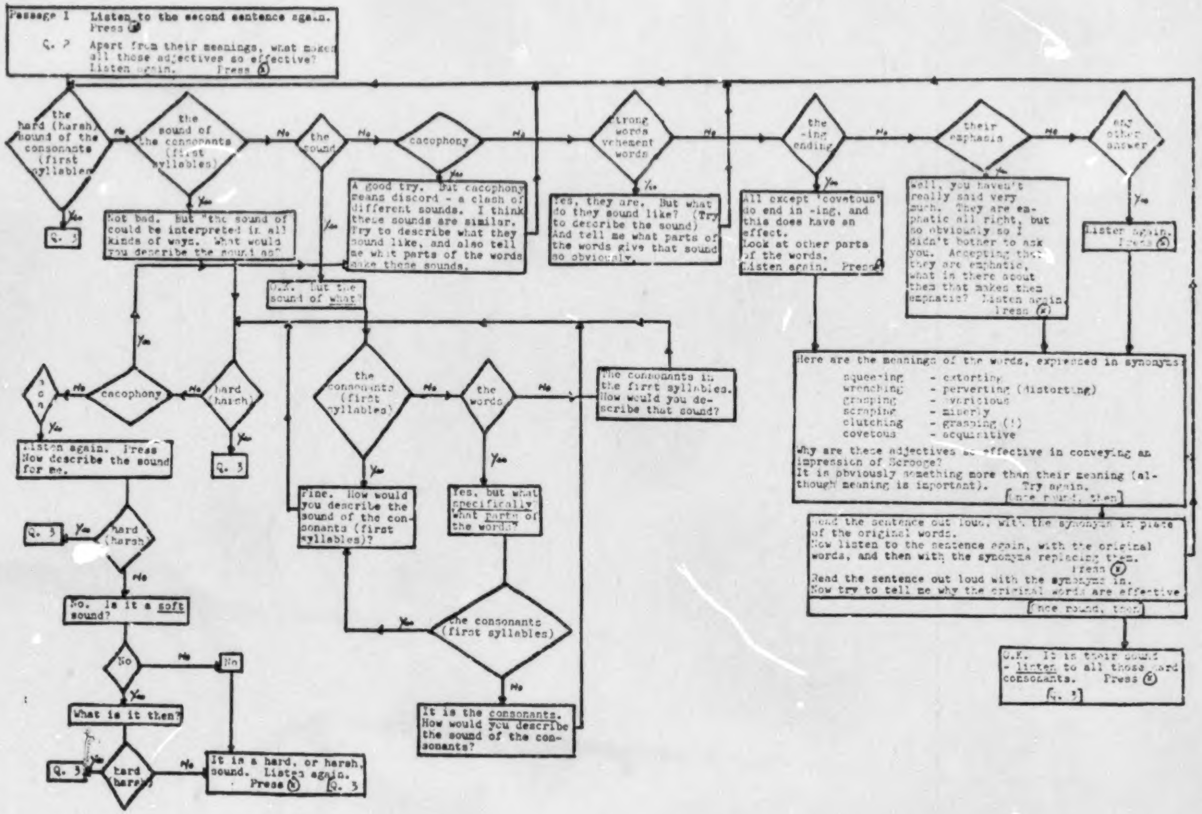
Good. [Q. 2

You are playing a game with me. It is a depressing rhythm. [Q. 2

Not bad. I doubt whether it tugs your heart strings; it is not really a weepy rhythm, though you may find it depressing. "Depressing" is the answer I was looking for. [Q. 2

Accept
1) Oh! But he was a tight-fisted hand at the grindstone
or 2) Oh! But he was a tight-fisted hand at the grindstone
or 3) Oh! But he was a tight-fisted hand at the grindstone
or 4) Oh! But he was a tight-fisted hand at the grindstone
or 5) Oh! But he was a tight-fisted hand at the grindstone.
No
Any other answer
Yes
Yes
depressing menacing
No
sad ugly gloomy
No
gives emphasis, gives impact, they are emphasised, any other answer
Yes
Yes
Yes
excites elates makes you feel glad makes you feel good makes you feel happy
No
Yes
Yes
1
No
5
No
Yes
1)sleepy 2)happy 4)numb
No
Yes
3)lachrymose
Yes

Figure III


the hard (harsh) sound of the consonants (first syllables)
the sound of the consonants (first syllables)
the sound
cacophony
strong words vehement words
the -ing ending
their emphasis
any other answer
the consonants (first syllables)
the words
the consonants (first syllables)
hard (harsh)
cacophony
hard (harsh)
No
hard (harsh)
Yes
No
Q. 3
No

Passage 1 Listen to the second sentence again. Press

Q. 2 Apart from their meanings, what makes all those adjectives so effective? Listen again. Press

Not bad. But "the sound of" could be interpreted in all kinds of ways. What would you describe the sound as?

O.K. But the sound of what?

A good try. But cacophony means discord - a clash of different sounds. I think these sounds are similar. Try to describe what they sound like, and also tell me what parts of the words make those sounds.

Yes, they are. But what do they sound like? (Try to describe the sound) And tell me what parts of the words give that sound so obviously.

All except 'covetous' do end in -ing, and this does have an effect. Look at other parts of the words. Listen again. Press

Well, you haven't really said very much. They are emphatic all right, but so obviously so I didn't bother to ask you. Accepting that they are emphatic, what is there about them that makes them emphatic? Listen again. Press

Listen again. Press

Here are the meanings of the words, expressed in synonyms:

squeezing - extorting
wrenching - perverting (distorting)
grasping - avaricious
scraping - miserly
clutching - grasping (!)
covetous - acquisitive

Why are these adjectives so effective in conveying an impression of Scrooge? It is obviously something more than their meaning (although meaning is important). Try again.

[Once round, then]

Read the sentence out loud, with the synonyms in place of the original words. Now listen to the sentence again, with the original words, and then with the synonyms replacing them. Press

Read the sentence out loud with the synonyms in. Now try to tell me why the original words are effective

[Once round, then]

O.K. It is their sound - listen to all those hard consonants. Press

[Q. 3]

The consonants in the first syllables. How would you describe that sound?

Fine. How would you describe the sound of the consonants (first syllables)?

Yes, but what specifically? What parts of the words?

It is the consonants. How would you describe the sound of the consonants?

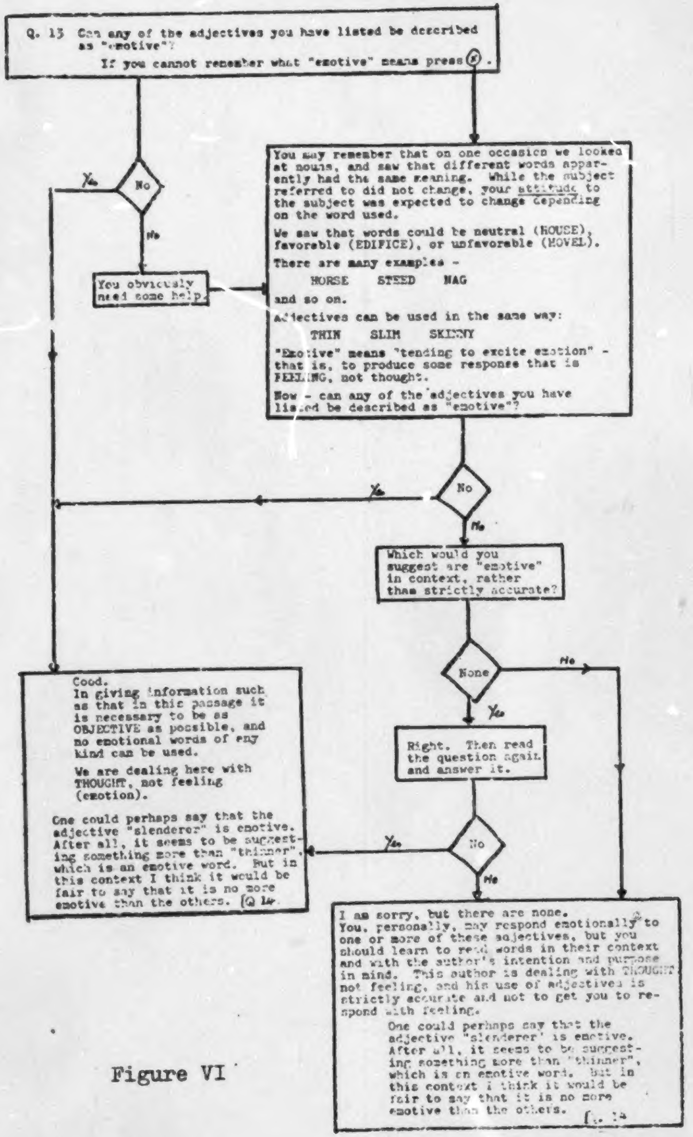
Listen again. Press Now describe the sound for me.

No. Is it a soft sound?

What is it then?

It is a hard, or harsh, sound. Listen again. Press Q. 3

Figure IV



Q. 13 Can any of the adjectives you have listed be described as "emotive"?

If you cannot remember what "emotive" means press ⓧ.

You obviously need some help.

You may remember that on one occasion we looked at nouns, and saw that different words apparently had the same meaning. While the subject referred to did not change, your attitude to the subject was expected to change depending on the word used.

We saw that words could be neutral (HOUSE), favorable (EDIFICE), or unfavorable (HOVEL).

There are many examples -

HORSE STEED NAG

and so on.

Adjectives can be used in the same way:

THIN SLIM SKINNY

"Emotive" means "tending to excite emotion" - that is, to produce some response that is FEELING, not thought.

Now - can any of the adjectives you have listed be described as "emotive"?

Which would you suggest are "emotive" in context, rather than strictly accurate?

Right. Then read the question again and answer it.

Good.
In giving information such as that in this passage it is necessary to be as OBJECTIVE as possible, and no emotional words of any kind can be used.

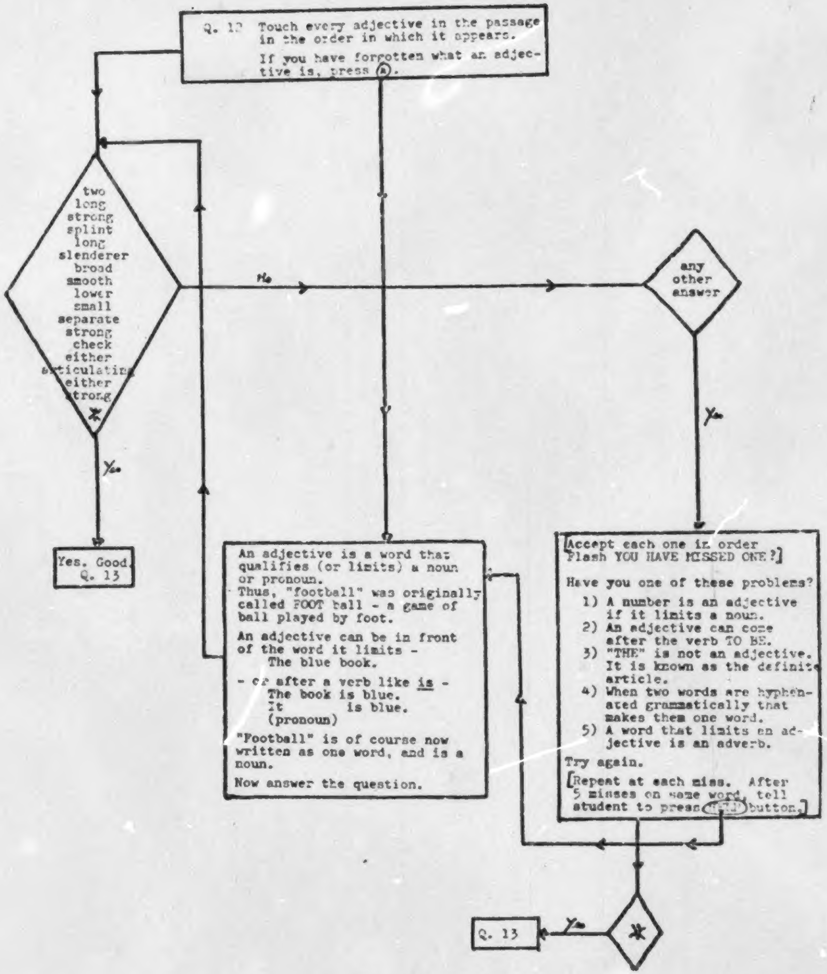
We are dealing here with THOUGHT, not feeling (emotion).

One could perhaps say that the adjective "slenderer" is emotive. After all, it seems to be suggesting something more than "thinner", which is an emotive word. But in this context I think it would be fair to say that it is no more emotive than the others. [Q 14

I am sorry, but there are none.
You, personally, may respond emotionally to one or more of these adjectives, but you should learn to read words in their context and with the author's intention and purpose in mind. This author is dealing with THOUGHT not feeling, and his use of adjectives is strictly accurate and not to get you to respond with feeling.

One could perhaps say that the adjective "slenderer" is emotive. After all, it seems to be suggesting something more than "thinner", which is an emotive word. But in this context I think it would be fair to say that it is no more emotive than the others. [Q. 14

Figure VI



Q. 12 Touch every adjective in the passage in the order in which it appears.

If you have forgotten what an adjective is, press ⓐ.

An adjective is a word that qualifies (or limits) a noun or pronoun.
Thus, "football" was originally called FOOT ball - a game of ball played by foot.

An adjective can be in front of the word it limits -
The blue book.

- or after a verb like is -
The book is blue.
It is blue.
(pronoun)

"Football" is of course now written as one word, and is a noun.

Now answer the question.

[Accept each one in order
Flash YOU HAVE MISSED ONE ?]

Have you one of these problems?

1) A number is an adjective if it limits a noun.
2) An adjective can come after the verb TO BE.
3) "THE" is not an adjective. It is known as the definite article.
4) When two words are hyphenated grammatically that makes them one word.
5) A word that limits an adjective is an adverb.

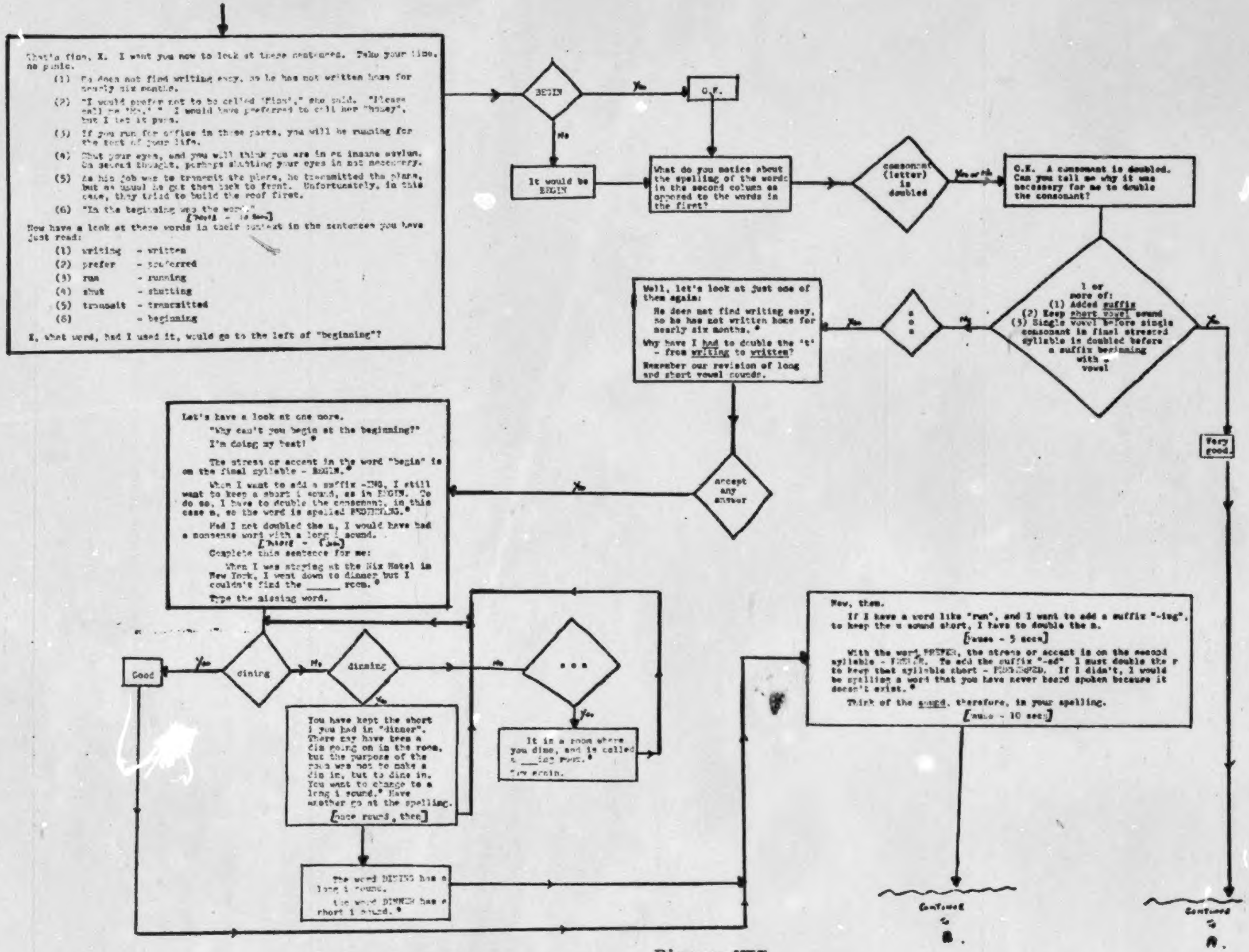
Try again.

[Repeat at each miss. After 5 misses on same word, tell student to press HELP button.]

Yes. Good. Q. 13

Q. 13

Figure V


BEGIN
Yes
No
O.K.
It would be BEGIN
consonant (letter) is doubled
Yes or ok
1 or more of: (1) Added suffix (2) Keep short vowel sound (3) Single vowel before single consonant in final stressed syllable is doubled before a suffix beginning with a vowel
. . .
accept any answer
Very good.
dining
dinning
. . .
Good
Continued to B.
Continued to A.

That's fine, X. I want you now to look at these sentences. Take your time, no panic.

(1) He does not find writing easy, so he has not written home for nearly six months.
(2) "I would prefer not to be called 'Miss'," she said. "Please call me 'Ms.' " I would have preferred to call her "honey", but I let it pass.
(3) If you run for office in these parts, you will be running for the rest of your life.
(4) Shut your eyes, and you will think you are in an insane asylum. On second thought, perhaps shutting your eyes is not necessary.
(5) As his job was to transmit the plans, he transmitted the plans, but as usual he got them back to front. Unfortunately, in this case, they tried to build the roof first.
(6) "In the beginning was the word."

[Pause - 10 secs]

Now have a look at these words in their context in the sentences you have just read:

(1) writing - written
(2) prefer - preferred
(3) run - running
(4) shut - shutting
(5) transmit - transmitted
(6) - beginning

X, what word, had I used it, would go to the left of "beginning"?

What do you notice about the spelling of the words in the second column as opposed to the words in the first?

O.K. A consonant is doubled. Can you tell me why it was necessary for me to double the consonant?

Well, let's look at just one of them again:

He does not find writing easy, so he has not written home for nearly six months.*

Why have I had to double the 't' - from writing to written?

Remember our revision of long and short vowel sounds.

Let's have a look at one more.

"Why can't you begin at the beginning?"

I'm doing my best!*

The stress or accent in the word "begin" is on the final syllable - BEGIN.*

When I want to add a suffix -ING, I still want to keep a short i sound, as in BEGIN. To do so, I have to double the consonant, in this case n, so the word is spelled BEGINNING.*

Had I not doubled the n, I would have had a nonsense word with a long i sound.

[Pause - 5 sec]

Complete this sentence for me:

When I was staying at the Nix Hotel in New York, I went down to dinner but I couldn't find the ______ room.*

Type the missing word.

You have kept the short i you had in "dinner". There may have been a din going on in the room, but the purpose of the room was not to make a din in, but to dine in. You want to change to a long i sound.* Have another go at the spelling.

[pace round, then]

It is a room where you dine, and is called a ___ing room.*

Try again.

The word DINING has a long i sound.

The word DINNER has a short i sound.*

Now, then.

If I have a word like "run", and I want to add a suffix "-ing", to keep the u sound short, I have to double the n.

[Pause - 5 secs]

With the word PREFER, the stress or accent is on the second syllable - PREFER. To add the suffix "-ed" I must double the r to keep that syllable short - PREFERRED. If I didn't, I would be spelling a word that you have never heard spoken because it doesn't exist.*

Think of the sound, therefore, in your spelling.

[Pause - 10 secs]

Figure VII

A COMPUTER MANAGEMENT SUPPORT SYSTEM FOR INDIVIDUALLY GUIDED EDUCATION

Mary K. Cooper and Gary R. Tobey
PORTLAND PUBLIC SCHOOLS

ABSTRACT

In 1973, Llewellyn Elementary School in Portland, Oregon, became a pilot school for Individually Guided Education (IGE). The management of the system was originally via a manual process. As the teachers added more subject areas to their individualized program, it became apparent that the management system would have to be improved if IGE was to continue to be viable. A computer-based management support system was developed and implemented early in 1975. This has resulted in a savings of approximately 350 hours of manual, clerical effort each year. In addition to satisfying all of the reporting requirements of the prior manual system, the computer system provides many new management reports which assist the school staff in planning, evaluation, and control of the IGE program. The development cost, to date, for the computerized system is approximately $10,000.

INTRODUCTION

In recent years, much has been said and written regarding Computer Managed Instruction (CMI). This paper describes a relatively simple but effective computer-based system that was developed to assist with the instructional management needs of a particular elementary school. The computer system operates in a "batch" mode on the School District's central computer. The system was expanded to include one more school in the fall of 1975. The School (Llewellyn Elementary) and the Individually Guided Education (IGE) program will be described along with the computer-based support system.

BACKGROUND

Llewellyn Elementary School is located in the Southeast part of the city. Long a traditional elementary school of approximately 450 students in grades K-8, Llewellyn is now in its third year of IGE implementation.

Several things necessitated this change. Once a stable middle class community, many families began the move out of the city — their homes rented to other families of widely varying socioeconomic backgrounds. Teachers found a wide difference in educational needs of the children now in their classes. For example, there would be several children of high achievement levels and many more at extremely low levels. Grade level materials and traditional approaches no longer worked. No one teacher could meet the needs of all.

In a search for an instructional program to meet these needs, administrators read of a program under development by the Wisconsin Research and Development Center for Cognitive Learning at the University of Wisconsin (Madison), called IGE -- INDIVIDUALLY GUIDED EDUCATION. They visited Madison, observed the program in operation, and returned confident that it would work for Llewellyn as well.

What is IGE?

First, it is a PHILOSOPHY -- a philosophy that holds that each child is unique and develops at his own rate. Likewise, children learn at different rates and through different methods. All children can and will learn when provided with the proper instruction at the proper time.

Basic to IGE is a PATTERN of ORGANIZATION for instruction -- the Multiunit school. At the classroom level, the non-graded unit replaces the age-graded, self-contained classroom. Each unit includes a unit leader, two to four other teachers, an aide, and 80-130 students. Children in a unit have a three to four-year age span.

At the heart of IGE is a model of instructional PROGRAMMING for the individual student. Ideally, each child's program is based on how and at what pace he learns best and where he stands on mastering specific skills. Attempting to achieve this ideal involves a series of steps which are outlined in Figure 1.

The instructional program begins with the identification of specific instructional objectives that are appropriate for the children of a given unit. A criterion-referenced test is then used to assess each child's level of mastery for each skill. The children are then placed into small instructional groups based on a skill that they need. Instruction is provided for a 2 to 3 week period which is followed by a post-testing to determine attainment of the objective. If a child has reached the mastery level (usually an 80 percent criterion is set) on the post-test, he is placed in another skill grouping. If he does not master the skill, he will be grouped again in the skill at another time and with another member of the instructional staff.

The Llewellyn school began its use of the instructional model with a reading management program developed at the Wisconsin Research and Development Center -- The Wisconsin Design for Reading Skills Development. The Design is organized into six skill areas: Word Attack, Study Skills, Comprehension. Self-Directed Reading, Interpretive Reading, and Creative Reading (See Figure 2). Materials for the first three areas include lists of skills with related behavioral objectives and criterion-referenced tests for assessing children's mastery of these skills. Resource and management procedures for teachers are supplied. Grouping of children in instructional groups and the record keeping of skills mastered is by means of an edge punched card-sorting system in which the data for a given skill area is kept on a profile card for each child. To identify pupils with common skill needs, the teacher selects a given skill, passes a needle through the appropriate hole in a stack of cards and shakes off the loose cards. The cards remaining on the needle belong to pupils who need to work on that skill.

This method works efficiently when only one skill area (i.e., Word Attack) is involved. But as a teaching team begins providing instruction in Study Skills and Comprehension and adapts the model to an objective based mathematics program -- many hours are spent in card sorting and in record keeping.

It was at this point in the program that it became apparent that some improved method of overall record keeping and of selecting and grouping students was mandatory. The School District's Data Processing Department was contacted and the problem was related to them. The idea of managing instructional grouping and record keeping was intriguing, and together, Unit Leaders and Data Processing personnel designed the system.

THE COMPUTER MANAGEMENT SYSTEM

During the System Survey stage of the project, it was decided that the system would be developed by the Data Processing Department as a "pilot project". This meant a relatively low level of resources would be expended on the project, certain steps in the standard "System Approach" normally followed, could be eliminated, and the system would be designed primarily to fit the needs of the specific school as opposed to developing a generalized system for many. It also meant that Data Processing could get a basic system up quickly, and then continue to improve, refine, and tailor it as needed during the early implementation stages. Several key points that were to be followed in the system development were identified. Among them were:

1. The school personnel would dictate the needs of the system, and it would be designed specifically for them. The existing manual process would be changed only when absolutely necessary.

2. The system was to be kept simple with a minimum of resource expenditure. Yet, it was recognized that the Llewellyn system might be expanded to other schools at a later time, and thus, this would be considered in the design.

3. The computer was to assist the Teacher or Unit Leader, not take over.

4. Computer assistance was to be provided only where the benefits were greatest as opposed to developing an overall system. For example, it was decided not to score the tests, but merely to input the test scores to the computer. Some factors which resulted in the decision to not score tests were:

   (a) *Testing can be written, oral or by observation.*

   (b) *Tests are short, easy to score, and the teacher can detect special problem areas.*

   (c) *Test scoring by computer would have required faster turnaround than could be easily provided by a "batch" system.*

   (d) *The Data Processing Department did not have a scanner capable of scoring tests in-house.*

5. The main decision processes were to be left to the Teachers and Unit Leaders. The system was only to suggest and provide pertinent information.

The Wisconsin Research and Development Center was contacted early in the study to see if they were planning a computer-based support system. It was found that they were, but it was in the early stages of development and would need to be modified to meet our specific needs and computer hardware. They did, however, provide some excellent information related to their planned system.

The basic system can best be described by referring to the general flow model (See Figure 3). There are several special reports and input documents in addition to those indicated in Figure 3. They are required primarily for system and file initialization and maintenance, and will not be described in detail here.

The normal process begins with a determination by the Unit Leaders of which skill groups they want to teach in the next grouping cycle. They make this determination based on Unit Performance Profile reports (See Figure 4) and a knowledge of what resources (rooms, teachers, materials) will be available. The Unit Performance Profile shows pertinent information regarding the performance of each student, and of the unit as a whole for each skill objective.

A Grouping Request Form is then prepared and sent to Data Processing. The desired skill groups, along with any special prerequisites or forced groupings, are noted on the form. The Grouping Requests are input to the computer. The computer produces Specific Grouping Reports (See Figure 5) by determining each student's eligibility based on previous history of skills mastered, prerequisites and number of previous attempts. The Specific Grouping Report is a list of all those students that are eligible for the specific skill. The students are listed on the report in sequence by need. Students that have made prior unsuccessful attempts and are not eligible for many, or any other skill groups, appear first on the report. The Unit Leaders modify the report as they see fit, drawing lines through those students that they do not wish to have taught in a given group. The resultant report provides a "class list" or "roster" for the teacher that is in charge of the group. A report is also provided that shows students that are not eligible for any of the skill groupings.

The learning process, usually of two or three weeks, then takes place. Students are tested and mark sense score cards, which have been prepunched by the computer and printed by a card interpreter, are marked with the students' scores, and submitted to Data Processing. The computer then updates the students' records. New Unit Profile Reports are then provided for use by the Unit Leaders in determining the next set of potential skills to be taught.

Upon request, individual Student Profile reports are printed (See Figure 6). They provide the teacher with a complete list of all skills that have been attempted by the student. It is useful in evaluating and reporting individual student progress. It is especially useful for parent/teacher and student/teacher conferences, and thus is often requested near the end of a grading period.

An Administrative Summary Report (See Figure 7) can also be provided upon request. It is used primarily by the principal and provides a summary of the performance of each Unit in the school.

CONCLUSION

The Computer Based-Instructional Management Support System, as briefly described in this paper, has been successfully implemented in the pilot school and is being used on a limited basis at one other school. The project has demonstrated to School District management that the computer, operating in a batch mode, can provide a significant and cost effective contribution to the instructional process. The door has been opened for additional future developments in the area of direct instructional support by the computer.

REFERENCES

Belt, S.L., and Spuck, D.W. Computer Application In Individually Guided Education: A Computer Based System for Instructional Management (WIS-SIM) Needs and Specifications, Madison, Wisconsin: Wisconsin Research and Development Center for Cognitive Learning, 1974.

Education U.S.A. Special Report. Individually Guided Education and the Multiunit School. Arlington, Virginia: National School Public Relations Association, 1972.

Klausmeier, H.J., Quilling, M.R., and Sorenson, J.S. The Development and Evaluation of the Multiunit Elementary School, 1966-1970. Madison, Wisconsin: Wisconsin Research and Development Center for Cognitive Learning, Technical Report #158, 1971.

Klausmeier, H.J., Sorenson, J.S., and Quilling, M.R. Instructional Programming for the Individual Pupil in the Multiunit Elementary School. The Elementary School Journal, 1971, 72(2), 88-101.

MAJOR INTEREST GROUPS

WISCONSIN RESEARCH AND DEVELOPMENT CENTER
FOR COGNITIVE LEARNING
University of Wisconsin-Madison
1025 West Johnson Street
Madison, Wisconsin 53706

****

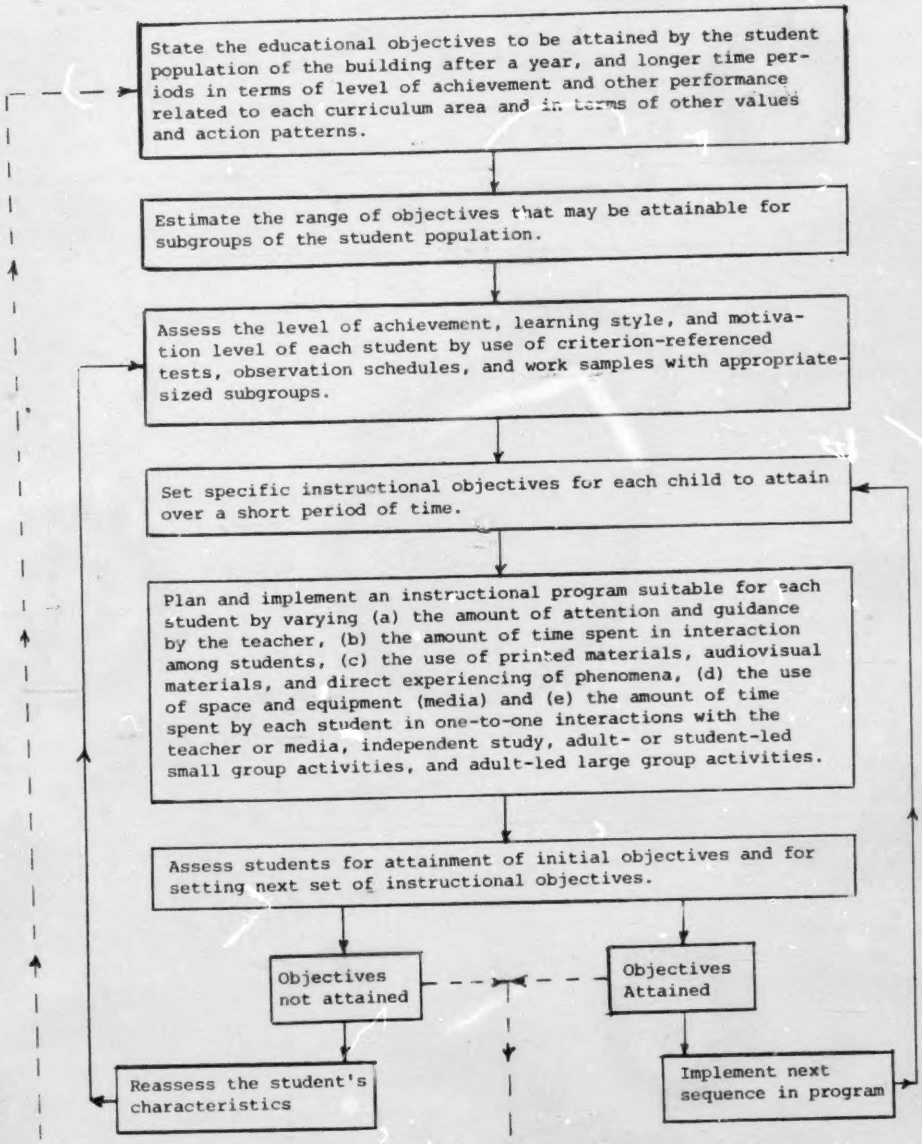
INSTITUTE FOR DEVELOPMENT OF EDUCATIONAL
ACTIVITIES
5335 Far Hills Avenue
Suite 300
Dayton, Ohio 45429

****

EXECUTIVE SECRETARY
ASSOCIATION FOR INDIVIDUALLY GUIDED EDUCATION
1025 West Johnson Street
Madison, Wisconsin 53706

****

THE UNIVERSITY OF WISCONSIN/SEARS-ROEBUCK
FOUNDATION PROJECT
1025 West Johnson Street
Madison, Wisconsin 53706


State the educational objectives to be attained by the student population of the building after a year, and longer time periods in terms of level of achievement and other performance related to each curriculum area and in terms of other values and action patterns.

Estimate the range of objectives that may be attainable for subgroups of the student population.

Assess the level of achievement, learning style, and motivation level of each student by use of criterion-referenced tests, observation schedules, and work samples with appropriate-sized subgroups.

Set specific instructional objectives for each child to attain over a short period of time.

Plan and implement an instructional program suitable for each student by varying (a) the amount of attention and guidance by the teacher, (b) the amount of time spent in interaction among students, (c) the use of printed materials, audiovisual materials, and direct experiencing of phenomena, (d) the use of space and equipment (media) and (e) the amount of time spent by each student in one-to-one interactions with the teacher or media, independent study, adult- or student-led small group activities, and adult-led large group activities.

Assess students for attainment of initial objectives and for setting next set of instructional objectives.

Objectives not attained
Objectives Attained
Reassess the student's characteristics
Implement next sequence in program
Feedback Loop

FIGURE 1: Instructional Programing Model in IGE

LLEWELLYN PRIMARY SCHOOL
(Sample Objectives)
READING SKILL SEQUENCE

Word Attack Skills: (K-4) 45 Objectives (linear sequence)

1. Shapes; Letters and Numbers
2. Rhyming Words and Phrases
3. Words and Phrases
4. Beginning and Ending Consonant Sounds; Consonant Blends
5. Plurals; Possessives
6. Long Vowel Sounds
7. Consant Digraphs
8. Vowel + r, a + l, a+w, etc.
9. Dipthongs
10. Base Words
11. Homonyms
12. Multiple Meanings
13. Silent Letters
14. Syllabication

Comprehension Skills: (1-6) 34 Objectives (linear sequence and concept strands)

1. Main Idea
2. Sequence
3. Reasoning
    a. Synthesizes information
    b. Predicts outcomes
    c. Cause and effect relationships
    d. Syllogisms
4. Detail
5. Context Clues
6. Affixes

Study Skills: (3-8) 71 Objectives (linear sequence and concept strands)

1. Maps
    a. Representation
    b. Orientation
    c. Measurement
2. Graphs and Tables
3. References
    a. Utilization
    b. Location
    c. Organization and evaluation

OTHER ASPECTS OF PROGRAM. Basal Reading Series

1. Self-Directed Reading
2. Interpretive Reading
e. Creative Reading

FIGURE 2

IGE/CMI PILOT PROJECT

GENERAL FLOW MODEL

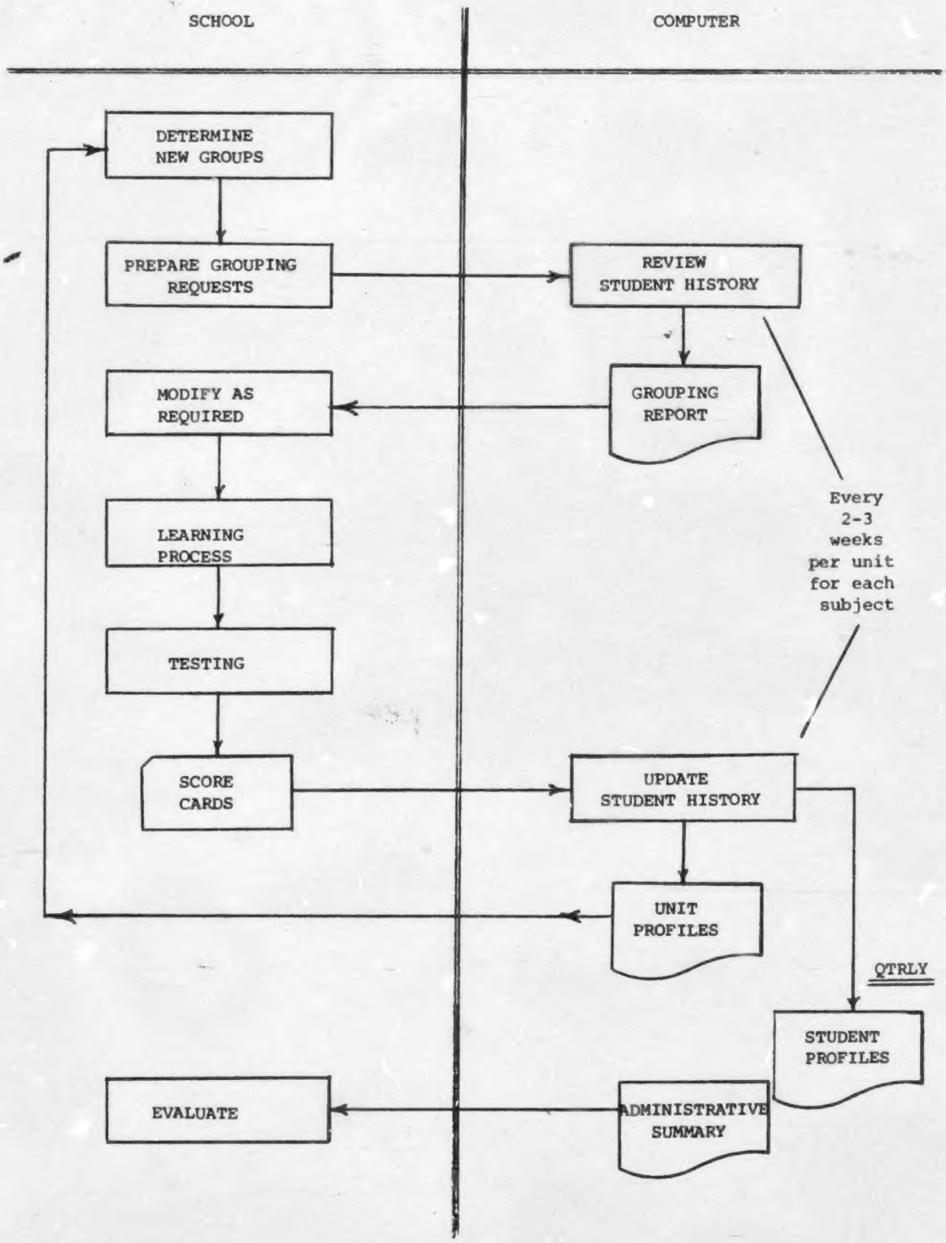

SCHOOL
COMPUTER
DETERMINE NEW GROUPS
PREPARE GROUPING REQUESTS
REVIEW STUDENT HISTORY
GROUPING REPORT
MODIFY AS REQUIRED
Every 2-3 weeks per unit for each subject
LEARNING PROCESS
TESTING
SCORE CARDS
UPDATE STUDENT HISTORY
UNIT PROFILES
QTRLY
STUDENT PROFILES
EVALUATE
ADMINISTRATIVE SUMMARY

FIGURE 3

467

```
600040-04 REV-035                                  UNIT PERFORMANCE PROFILE              DATE 04-11-75        PAGE  16
          SCHOOL LLEWELLYN          UNIT P                  CLUSTER = A01 THRU B13 - WORD ATTACK

STUDENT NAME                PRIOR
                            MASTERY      A01   A02   A03   A04   A05   A06   A07   B01   B02   B03   B04   B05   B06   B07   B08   B09   B10   B11   B12   B13

[redacted]NS PETER                      100M     M     M     M  100M     M     M           M           M     M        100M  100M
-------------------------------------------------------------------------------------------------------------------------------
[redacted]RTZ CHRIS J                      M     M     M     M     M     M     M     M     M     M     M     M           M           M     M                 M
[redacted]TI SCOTT J                       M     M     M     M     M     M     M           M     M     M  080M
[redacted]NS JAMES M                       M     M     M     M     M     M     M           M     M     M     M           M              087M           M     M
[redacted]ETON MARK L                      M     M     M     M     M     M     M           M           M  085M
[redacted]ON MIKE                          M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M  100M     M
[redacted]STEVEN                           M     M     M     M     M     M     M           M     M     M  100M     M  080M
-------------------------------------------------------------------------------------------------------------------------------
[redacted]R VINCENT C                      M     M  060      M     M     M     M           M           M     M        053   088M                             M
[redacted]ER WILLIAM DEAN                  M     M     M     M     M     M     M     M     M     M     M     M           M     M     M  093M     M     M     M
[redacted]CKER MICHELE L                   M     M     M     M     M     M     M           M     M     M  070      M                 M
[redacted]EUGENE U                         M     M  100M     M  100M     M     M           M     M     M     M     M  067   076
[redacted]VAN MICHAEL S                    M     M     M     M     M     M     M           M     M     M  090M
[redacted]SKY NYE A                     100M     M     M     M     M     M     M           M     M     M  100M     M        100M     M                       M
-------------------------------------------------------------------------------------------------------------------------------
[redacted]R L DOUGLAS                      M     M     M     M     M     M                 M     M     M  065
[redacted]HER NINA                         M     M     M     M     M     M     M           M     M     M     M           M     M     M  073
[redacted]EN STEVEN D                      M  067      M     M     M     M     M           M     M     M  090M              100M
[redacted]HAKIKO                           M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M                 M
[redacted]SON MICHAEL                      M     M  100M  100M     M     M     M           M              095M     M        082M
[redacted]END MELISSA                      M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M  100M     M     M     M
-------------------------------------------------------------------------------------------------------------------------------
[redacted]PAMELA B                         M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M  100M     M
[redacted]RS KAREN A                       M     M     M     M     M     M     M           M     M     M     M     M     M     M        073            M
[redacted]N CHERYL L                       M     M     M     M     M     M     M     M     M     M     M     M     M     M     M  100M  100M     M  092M     M
[redacted]ER JAMES G                       M     M     M     M     M     M     M           M     M     M     M     M  087M     M                 M     M     M
[redacted]ER JILL D                        M     M     M     M     M     M     M           M     M     M     M        100M  094M                             M
[redacted]ANDREW W                         M     M     M     M     M     M     M           M     M     M     M     M     M     M     M  073                  M
-------------------------------------------------------------------------------------------------------------------------------
[redacted]PATRICIA L                       M  087M     M     M     M     M     M           M              015
[redacted]JERROLD P                        M     M     M     M     M     M     M           M     M     M     M           M  100M        087M
[redacted]FF JAN                           M  073      M     M     M     M                 M     M     M                    053
[redacted]DEBORAH SUE                      M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M     M
[redacted]LAURA L S                        M     M     M     M     M     M     M     M     M     M     M     M           M           M  100M                 M
[redacted]LISA D                           M     M     M     M     M     M                 M              065               078
-------------------------------------------------------------------------------------------------------------------------------
[redacted]TARRILL A                     090M     M     M     M  100M     M     M           M           M     M     M  093M

TOTAL MASTERED                            80    77    80    80    80    81    77    20    79    66    72    68    44    51    42    29    28    19    25    31

TOTAL UNMASTERED                           1     3     1                                                     10           5     7           6
-------------------------------------------------------------------------------------------------------------------------------
TOTAL UNTRIED                              1     2     1     2     2     1     5    62     3    16    10     4    38    26    33    53    48    63    57    51
```

FIGURE 4: Unit Performance Profile

FIGURE 4: Unit Performance Profile

```
600040-02 REV-042   SCHOOL LLEWELLYN        (378)  SPECIFIC GROUPING REPORT       DATE 05/13/75
                          UNIT P                                                    PAGE     5
GROUPING FOR SKILL B10              OF WORD ATTACK                PREREQS:

STUDENT #     STUDENT NAME               GRADE   ATTEMPTS     LAST     OTHER GROUPINGS
                                                 NO. MONTH   PERCENT

A378000143    [redacted] MELISSA C         02      2    03      073
A3780001[illegible]    [redacted] INA               02      1    03      053     W-B11
A400002548    [redacted] R NINA            02      1    03      073     W-B11
A378000198    [redacted] PETER J           01      1    05      067     W-B09,W-B11
A378000199    [redacted] DNEY A            01      1    05      067     W-B09,W-B11
A296001095    [redacted] CHEAL E           02      1    03      047     W-B09,W-B11
A392000072    [redacted] TEVEN             01      1    05      047     W-B09,W-B11
A378000223    [redacted] ANTHONY W         01      0                    W-B09,W-B11
A378000158    [redacted] N DAVID           02      0                    W-B09,W-B11
A378000226    [redacted] BRENDA L          01      0                    W-B09,W-B11
A378000140    [redacted] MARI K            02      0                    W-B09,W-B11
A128000418    [redacted] JILL D            02      0                    W-B09,W-B11
A378000204    [redacted] DAVID C           01      1    05      073     W-B05,W-B07,W-B11
A378000212    [redacted] UGENE U           01      1    05      067     W-B07,W-B09,W-B11
A400002567    [redacted] N MICHAEL         01      1    05      067     W-B07,W-B09,W-B11
A392000050    [redacted] I ROSS A          01      0                    W-B07,W-B09,W-B11
A378000196    [redacted] ARK S             01      0                    W-B07,W-B09,W-B11
A400002587    [redacted] HEIDI K           01      0                    W-B07,W-B09,W-B11
A378000218    [redacted] NANCY M           01      0                    W-B07,W-B09,W-B11
A400002595    [redacted] JOHN A            01      0                    W-B05,W-B09,W-B11
A378000116    [redacted] ADLEY A           02      0                    W-B05,W-B09,W-B11
A378000220    [redacted] A M               01      0                    W-B05,W-B09,W-B11
A378000221    [redacted] WARREN N          01      0                    W-B05,W-B09,W-B11
A378000222    [redacted] JULIA P           01      0                    W-B05,W-B09,W-B11
A150001259    [redacted] MONTE             02      0                    W-B05,W-B09,W-B11
A352000278    [redacted] EATHER LESLIE     02      0                    W-B07,W-B09,W-B11
A378000132    [redacted] N ROBERT A        02      0                    W-B07,W-B09,W-B11
A378000208    [redacted] ONALD J           01      0                    W-B05,W-B09,W-B11
A378000210    [redacted] SCOTT J           01      0                    W-B07,W-B09,W-B11
A398000024    [redacted] ON MARK L         01      0                    W-B07,W-B09,W-B11
A378000164    [redacted] VINCENT C         02      0                    W-B07,W-B09,W-B11
A378000163    [redacted] N MICHAEL S       02      0                    W-B07,W-B09,W-B11
A378000234    [redacted] L DOUGLAS         01      0                    W-B05,W-B09,W-B11
A378000235    [redacted] STEVEN D          01      0                    W-B07,W-B09,W-B11
A400002495    [redacted] F JAN             01      0                    W-B05,W-B09,W-B11
A378000217    [redacted] ISA D             01      0                    W-B05,W-B09,W-B11
```

FIGURE 5: Specific Grouping Report

FIGURE 5: Specific Grouping Report

```
600040-03 REV-082                   75/76    STUDENT PROFILE                    DATE 02-25-76
    FOR [redacted]ER KELLY J        SCHOOL LLEWELLYN         UNIT I                PAGE     4

SUBJECT                           --- QTR 1 ----  --- QTR 2 ----  --- QTR 3 ----  --- QTR 4 ----
 SKILL TITLE           ATTEMPTS -> 1    2    3     1    2    3     1    2    3     1    2    3


COMPREHENSION   MASTERED

 PICTURE TOPICS            A01    ...M
 FIRST OR LAST EVENT       A02    ...M
 SYNTHESIS                 A03    ...M
 OUTCOMES                  A04    ...M
 PARAGRAPH TOPICS          B01                    073  093M
 OUTCOMES                  B03                                    083M
 DETAIL                    B04    ...M

STUDY SKILLS    UNMASTERED

 MEASUREMENT: SIZE         C04                                    040
 MEASUREMENT: DISTANCE     C05                                    053

STUDY SKILLS    MASTERED

 PT--REPRESENTATION        A01    ...M
 POSITIONS OF OBJECTS      A02    ...M
 MEASUREMENT: SIZE         A03    ...M
 MEASURMENT: DISTANCE      B03                                    100M
 GRAPHS: RELATIVE AMOUNTS  B04                    087M

WORD ATTACK     MASTERED

 RHYMING WORDS             A01    ...M
 RHYMING PHRASES           A02    ...M
 SHAPES                    A03    ...M
 LETTERS NUMBERS           A04    ...M
 WORDS PHRASES             A05    ...M
 COLORS                    A06    ...M
 INITIAL CONSONANTS        A07    ...M
 SIGHT VOCABULARY          B01    ...M
 LEFT-RIGHT SEQUENCE       B02    ...M
 BEGINNING CONSONANTS      B03    ...M
 ENDING CONSONANTS         B04    ...M
 CONSONANT BLENDS          B05    ...M
 SHORT VOWELS              B06    100M
```

FIGURE 6: Student Profile

FIGURE 6: Student Profile

```
600040-07 REV-035              ADMINISTRATIVE SUMMARY REPORT              DATE 04-11-75

                           - - - - Y T D   T O T A L S - - - - - - -    RATE OF ACQUISITION
  UNIT     # STUDENTS     # SKILLS      # ATTEMPTS      # ACQUIRED     (ACQUIRED/ATTEMPTS)
          PARTICIPATING   ATTEMPTED      (TOTAL)         (TOTAL)

   P           82            56            631             531                84.2 %

   J           91            57            799             731                91.5 %

   I            0             0              0               0                 0.0 %

   U            0             0              0               0                 0.0 %

 TOTAL        173           113           1430            1262                88.3 %



                           - - - - Y T D   A V E R A G E S - - - -
                           # SKILLS      # ATTEMPTS      # ACQUIRED

   P                          0.7           7.7             6.5

   J                          0.6           8.8             8.0

   I                          0.0           0.0             0.0

   U                          0.0           0.0             0.0

 TOTAL                        0.7           8.3             7.3
```

FIGURE 7: Administrative Summary Report

FIGURE 7: Administrative Summary Report

THE INTEGRATED USE OF PROGRAMMED INSTRUCTION AND CAI TO TEACH THE APPLICATION OF LOGIC AS A MEANS FOR DETERMINING THE VALUE OF MANAGEMENT INFORMATION - AN EXAMPLE OF THE USE OF CAI WITHIN THE DEPARTMENT OF DEFENSE COMPUTER INSTITUTE

by Robert H. Orr and Judith A. Watson
Department of Defense Computer Institute
Washington Navy Yard, Washington, D.C. 20374

ABSTRACT: The Department of Defense Computer Institute has developed an integrated approach to teaching functional managers and users how to get more out of their Computer Based Information Systems. This approach incorporates Programmed Instruction (PI), Standard Lectures, Seminar/Laboratory Sessions and Computer Assisted Instruction (CAI) Techniques. Specifically, the PI text instructs the student in the fundamentals of symbolic logic, emphasizing the notions of conjunction, disjunction and negation. Additionally, it serves as a foundation for a CAI lesson which presents the application of Boolean and Tri-state Logic concepts to aid in determining the value of management information.

The Department of Defense Computer Institute (DODCI) was established in 1964 for the expressed purpose of teaching both military and civilian managers, within the Department of Defense, how to better use the computer as a management tool. All of the Institute's courses range in length from two to ten days, and deal with such subjects as computer system security and privacy, computer performance evaluation, specifications for selection, systems analysis and design, information management, requirements for teleprocessing and a basic introduction to computer technology. It is in the latter course that the use of Computer Assisted Instruction (CAI) appears the most promising.

In considering how to best incorporate CAI techniques into DODCI's program of instruction, the following facts and circumstances were considered:

a. For a variety of reasons not germane to this discussion, the present instructional format consisting of lecture, time sharing laboratory and seminar techniques must be maintained.

b. DODCI's time sharing system consists of the Hewlett-Packard 2000C' computer system and twenty remote terminals servicing a normal course enrollment of forty students.

c. A maximum of two classroom hours can be devoted to CAI, although students may use the time sharing system during lunch and after the normal academic day.

Consequently, the philosophy that students should be able to complete all CAI lessons within twenty minutes has been adopted. This will maximize the students' opportunities to make use of the CAI lessons both during the allotted course time and the students' free time.

The remainder of this paper will be organized into two sections; the first describes the specific approach taken to adapt CAI principles to DODCI's needs, while the second describes an actual lesson.

Each CAI lesson is organized into three phases: an introductory phase, a reinforcement phase consisting of review sessions (i.e., review quizzes and remedial units), and a phase presenting new material consisting of learning units and a final test (See Figure 1). The total package provides learning and reinforcement to the student, as well as information to educators on the effectiveness of prior instruction.

The introductory phase gives the student an overview of the total lesson. This phase serves to place the student as much at ease as possible by explaining the interactive capabilities of computer terminals and familiarizing the student with the forthcoming lesson.

The reinforcement phase is designed to insure that the student has mastered the material previously taught through lecture, upon which the new subject matter is based. Therefore, each lesson author first determines the specific lectures to serve as foundations for the CAI lesson. The author then identifies those specific teaching points which should be mastered. The student's mastery is ascertained by administering a computer assisted test, the aforementioned review quiz; bear in mind that the subject matter covered in each review quiz pertains to a

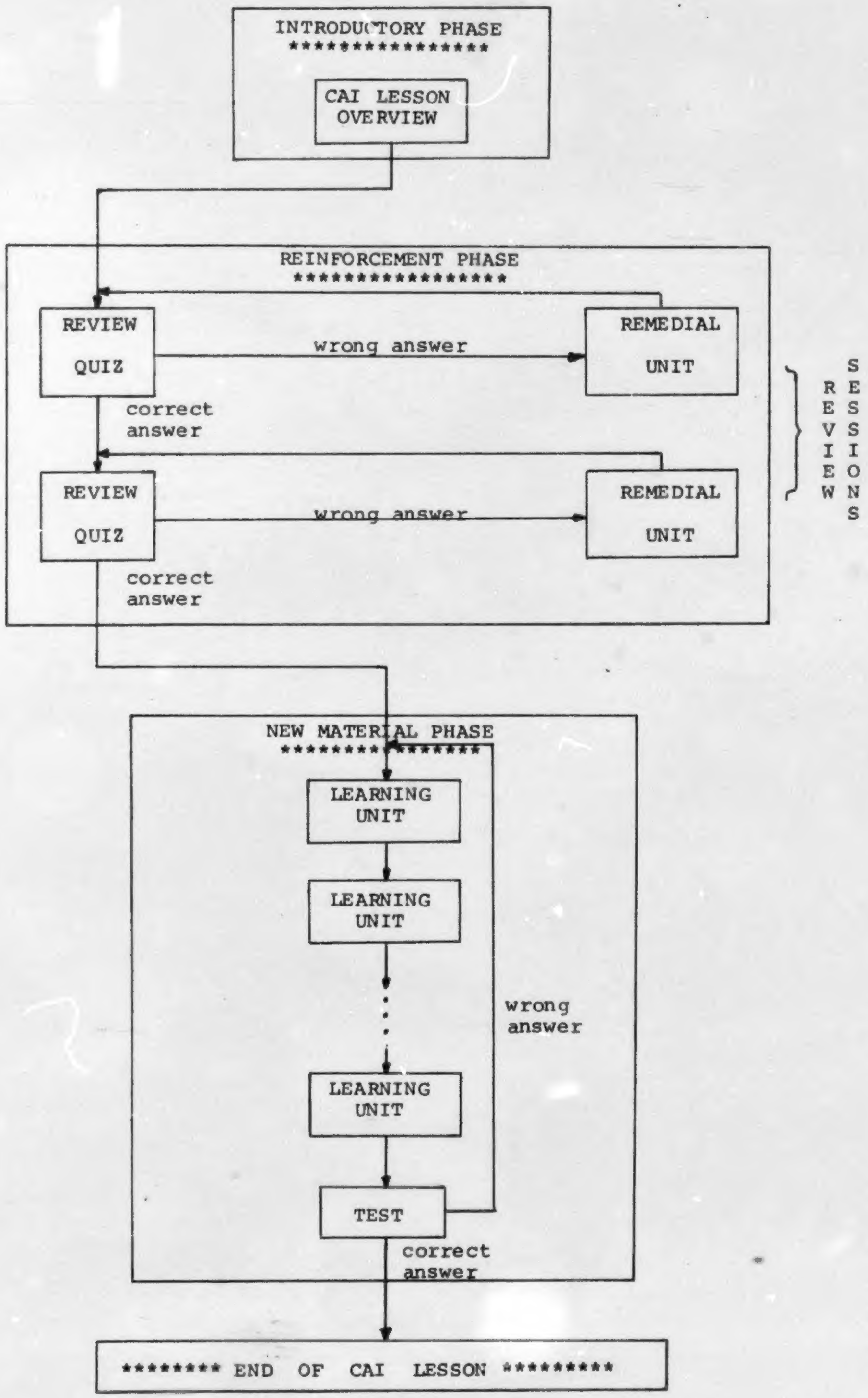

INTRODUCTORY PHASE
CAI LESSON OVERVIEW
REINFORCEMENT PHASE
REVIEW QUIZ
wrong answer
REMEDIAL UNIT
correct answer
REVIEW QUIZ
wrong answer
REMEDIAL UNIT
correct answer
REVIEW SESSIONS
NEW MATERIAL PHASE
LEARNING UNIT
LEARNING UNIT
LEARNING UNIT
wrong answer
TEST
correct answer
END OF CAI LESSON

Figure 1.

single course lecture. The quiz usually consists of five to seven questions. For each question, the student has one trial in which to answer correctly; he progresses until he renders an incorrect response, at which time the student is branched to a remedial unit. The remedial unit, activated contingent to prior student responses, is designed to reteach the material covered in the review quiz. Upon completion of the remedial unit, the student retakes the quiz to reassess his mastery. Correct responses on any re-test would still indicate mastery since the review quiz is confined to single word or multiple choice formats, and an incorrect response is not explained during the quiz. In the event a student again renders an incorrect answer, he is immediately returned to the remedial unit for further work. The possibility of this cycle proceeding indefinitely is not anticipated as the quiz has been designed to realize a high percentage of correct responses. Due to self-imposed limitations, the reinforcement phase is constrained to a maximum of two review quizzes, along with their associated remedial units. Once the student successfully completes these review sessions, he is branched to the new material portion of the lesson.

The new material phase reverses the sequence of events which transpire in the reinforcement phase. New subject matter is presented to the student in an interactive fashion. When completed, the student is tested to determine his mastery of the new subject. Any incorrect response will automatically result in the student's return to the beginning of this last phase. Correspondingly, all correct responses reflect sufficient comprehension of the material and thus, constitute successful lesson completion.

Consider an actual lesson illustrating the design philosophy described above. This lesson is based on the students' mastery of three concepts: elementary logic, mathematical modeling, and the value of management information.

The student learns the fundamentals of logic by completing a Programmed Instruction (PI) Text. The use of this PI text serves two purposes. First, it exposes the student to a different technique for the teaching of new material: this technique capitalizes on the benefits of self-paced instruction as a realistic and highly successful form of education. Second, it introduces the student to some fundamental CAI techniques (e.g. lesson frames, student dialogue, immediate reinforcement). This preliminary exposure should minimize any "culture shock" which might otherwise be experienced by a student encountering CAI for the first time.

The other concepts are introduced and first taught using the normal classroom lecture format. The three concepts must be mastered before the student is ready to attempt the new subject matter that comprises the final phase of the CAI lesson. A part of an overall Student Progress Review examination tests the students' comprehension of logic. The master of the other topics is ascertained as part of the reinforcement phase of the CAI lesson.

After completing the introductory phase, the student enters the first review session and receives a quiz on modeling. As described earlier, any incorrect response will cause the student to be branched to a remedial unit designed to teach/review the types of models, their purposes, and the significant advantages and disadvantages of modeling.

Likewise, the next review session begins with a quiz on the notion of information values; this quiz is also supported by a remedial unit designed to guarantee the student's mastery of this vital subject. (NOTE: For the benefit of those not familiar with this topic, a brief explanation is presented below).

A quantitative approach for determining the value of management information in a manner similar to that of a production shop is generally not feasible. The production shop determines a retail value for a service or commodity based on the fixed and variable costs of producing that service or commodity. The retail cost is chosen to enhance profit margins. Figure 2 graphically portrays why this technique is inappropriate for assigning a value to management information.

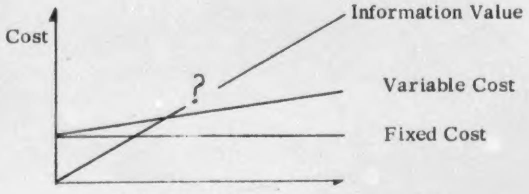


Figure 2

While the fixed cost for operating a management information system can be determined, and while the actual cost for producing information (variable cost) can be calculated, it is impossible to quantify the net worth of the information produced in order to justify its cost of production. Rather, managers must make a more subjective appraisal of information value. Consequently, students are taught a composite of various management theories referred to here as the "Value Added" approach. Essentially, information has its value enhanced as one or more of the following states are realized.

VALUE ADDED FACTORS

1. Administrators are given knowledge to match the scope of the problem for which they are responsible.

2. Decisions are made that otherwise would not be made.

3. Decisions by analysis prove better than decisions by insight.

4. The scarce commodity of managerial insight is freed from lower-level decisions and can be applied to higher-level problems.

5. Decisions are made sooner because of increased confidence.

Once the student has demonstrated his mastery of the aforementioned material, he is ready to begin the final phase of the CAI lesson. Here, the student is taught how to apply Boolean and tri-state logic functions to build a mathematical model which can be used in producing a relative measure of the value of management information. The solution of this model would provide the manager with a means for assessing the utility and effectiveness of his information system.

For example, two of the most important managerial resources are time and money. A series of Information Value measures can be defined by describing them first as functions of time and each of the five Value Added factors, and then, as functions of cost and the Value Added factors. As an illustration:

Let T, the time a manager spends reading and understanding some piece of information, be defined:

$$T = \begin{cases} 0 & \text{if time spent is "excessive", else} \\ 1 & \end{cases}$$

Let D, representing Value Added factor 5, be defined:

$$D = \begin{cases} 0 & \text{if decision is not made sooner, else} \\ 1 & \end{cases}$$

Next, define the first Information Value measure, IV_1 as a function of T and D:

$$\begin{array}{cc|cc} & & \multicolumn{2}{c}{D} \\ & & 0 & 1 \\ \hline \multirow{2}{*}{T} & 0 & 0 & 0 \\ & 1 & 0 & 1 \end{array} = IV_1$$

Here, the IV_i values can be interpreted as follows:

$$IV_i = \begin{cases} 1 & \text{if information is of "significant value", else} \\ 0 & \end{cases}$$

where $i = 1, 2, \ldots, n$
and n defines the number of Information Value measures to be modeled.

Repeating the above process, substitute time, T, with a variable representing cost.

Let C, the cost of preparing a piece of information, be defined:

$$C = \begin{cases} 1 & \text{if economical, else} \\ 0 & \end{cases}$$

Then IV_2 is defined as a function of C and D:

$$\begin{array}{cc|cc} & & \multicolumn{2}{c}{D} \\ & & 0 & 1 \\ \hline \multirow{2}{*}{C} & 0 & 0 & 0 \\ & 1 & 0 & 1 \end{array} = IV_2$$

Each of the other Value Added factors could be similarly defined, giving a total of ten different Information Value measures.

V, the value of the jth piece of information is defined as:

$$V_j = \frac{\sum_{i=1}^{n} IV_{ij}}{n}$$

For $j = 1, 2, \ldots, m$ pieces of information

$i = 1, 2, \ldots, n$ information value measures

Finally, a measure of Total Information Effectiveness, M, is defined:

$$M = \frac{\sum_{j=1}^{m} V_j}{m}$$

Note that in the above two matrices, the logical AND function has been duplicated. This need not have been the case. The CAI lesson emphasizes that each modeler must provide his own interpretation and must define each function in a way suitable to the modeler's analysis. The above definitions are the writers' interpretation. Recognize, also, that the CAI lesson emphasizes approach and technique rather than a detailed development of the entire model. Most of the function definitions are reserved for the student as exercises. Furthermore, the teaching philosophy is far less rigorous than the above definitions imply. This is because the students do not all possess the mathematical sophistication necessary to comprehend a formal description of the model. Hence, the teaching of this topic follows more intuitive lines.

Because the student may have some reservations concerning the precision provided by the Boolean functions, as a final teaching point, the use of tri-state variables is suggested. To illustrate, the functions T, D, and IV_1 are redefined below:

$$T = \begin{cases} -1 & \text{if time is excessive} \\ 0 & \text{if time is average, else} \\ +1 & \end{cases}$$

$$D = \begin{cases} -1 & \text{if not made sooner} \\ 0 & \text{if made slightly sooner, else} \\ +1 & \end{cases}$$

$$\begin{array}{cc|ccc} & & & D & \\ & & -1 & 0 & +1 \\ \hline & -1 & -1 & 0 & 0 \\ T & 0 & -1 & 0 & 0 \\ & +1 & -1 & +1 & +1 \end{array} = IV_1$$

The Information Value measure implies that the information itself is either irrelevant (-1), "nice-to-have" (0), or relevant (+1). The statistical measures previously defined are still valid.

Once these ideas have been presented, each student takes a final test to ascertain his mastery of the new material. Again, the student is automatically branched to the beginning of this final phase whenever an incorrect answer is given. Perfect completion of the test constitutes CAI lesson completion.

In summary, the Department of Defense Computer Institute is developing an integrated approach to its courses of instruction by combining the best features of lecture, Programmed Instruction, seminar/laboratory, and Computer Assisted Instruction to produce a well-balanced, highly effective curriculum.

Acknowledgements: All CAI programming was done using Hewlett-Packard's Instructional Dialogue Facility (IDF). The authors wish to thank our Director, Captain J. W. Riehl, USN for his many helpful suggestions. We also extend our thanks to Captain Clifton O. Reynolds, USA who programmed the bulk of this project.

AN ALTERNATIVE METHOD
FOR
TEACHING A CAI AUTHOR LANGUAGE

by

William R. Johnson
Director of Computer Resources
Loras College
Dubuque, IA 52001

ABSTRACT: Recent literature indicates that the CAI discipline is beginning to impact students at all levels of education, rather than remaining strictly as a research device. As CAI begins to impact students, the problem of communicating CAI principles and languages to the "average" teacher develops. An alternative method to the traditional approach of classroom instruction is to have the computer teach its own CAI language. In other words, to have a CAI tutorial created to teach a CAI language. An attempt to fulfill this goal has been accomplished by the development of a CAI tutorial to teach the CAI author language IDF. The main purpose of the tutorial was to provide instruction in IDF so as to effectively eliminate the initial teaching of IDF in the classroom. The main instructional goal of the tutorial was that the user would be able to create a CAI program using IDF with the author's manual functioning as a reference. The tutorial discussed eight major areas of IDF and followed a linear instructional strategy. A pre-test/post-test evalution procedure was used to determine the effectiveness of the tutorial. Results were significant at the .05 level, however, they also indicated some revision was necessary. It was determined that the major instructional goal had been achieved with relative success. In developing tutorials like the one indicated above, teachers need only go to the computer in their own school environment to learn its CAI language. In this manner, we can effectively eliminate the teaching of CAI languages in the classroom, and thus convey the principles of CAI more readily.

Introduction and Rationale

Large amounts of money and time have been spent in order to place computers in education, "... but the average teacher remains uninformed as to what it is and how it works."(9) The "average" teacher of whom Porter speaks needs to become knowledgeable about computers since the computer is becoming a part of everyday life in the school. Wallace has said, "Despite the surge of interest in CAI, there is a clear shortage of CAI material ... A key reason for this shortage is what might be called the teacher/programmer gap."(14) in order that this shortage may be lessened, a method or methods of training for the average teacher must be developed. Material must be developed to help educators learn procedures for organizing the content of a CAI program and understanding the basic concepts of CAI and the computer in general. This is accomplished to some degree in the CAI and computers in education courses offered on many campuses. These courses are excellent methods of acquainting the teacher with general computing concepts and the different aspects of CAI, however, many times they do not provide a firm foundation of programming techniques the teacher will need to use to "input" their modules onto the computer. It is essential that this be expanded to include the programming techniques and concepts so educators may become more aware of the capabilities of the particular computer available to them in their own school environment.

One of the possible vehicles for accomplishing this is to have the computer teach it's own CIA language, that is, to have a CAI tutorial written to teach a CAI language. Thus, these modules could be included as a part of the basic software package given with each computer. This would then essentially eliminate the responsibility of teaching the programming concepts in the classroom.

The major purpose of this study was to work toward the above objective by developing a CAI tutorial designed to teach a CAI language. The subject area of the tutorial was the fundamentals of the Hewlett-Packard (HP) CAI language called Instructional Dialogue Facility (IDF).

The tutorial this paper discusses is the first in a series of three. The second tutorial will deal with the utilization of the IDF Editor-Documentor (EDDOC) program and the third, the use of the IDF Utility Program (IDFUTI). The second and third modules are currently in the pre-computer stages of development.

There are several reasons for the development of the tutorial. It is hoped that the tutorial will help the "average" teacher keep pace with the rapid development of computer technology. Also, it is hoped the tutorial will help in overcoming the "teacher/programmer gap" (14) and thus, as a consequence, help to eliminate the shortage of CAI material. The tutorial is also an attempt to show the computer illiterate teacher that, when used properly, it can be an extremely valuable tool, giving him/her more time to develop one-to-one relationships with each student. It is further hoped that the tutorial will be a "stepping stone" for teachers to gain insight into the principles of CAI and more generally the application of computers in education.

The main purpose of the tutorial is to provide instruction in IDF so as to effectively eliminate the initial teaching of IDF in the classroom. After completing the tutorial, the user may then attend a class or several classes or seek a consultant to answer questions and learn some advanced techniques of IDF not discussed in the tutorial.

The target population for which this tutorial was developed is that group of educators that have had little or no previous computer programming experience. It can be used for both pre-service and in-service training.

The major instructional goal of the tutorial is to provide the basic understanding needed to create a CAI program using IDF, with the author's manual functioning as a reference. In addition, the module should give the user an "appreciation" of the limitations and enhancements a computer can offer education. Finally, it should give the user an understanding of the process involved in the creation of computer based instructional material on the HP computer.

Design

In order for the major instructional goal to be accomplished, it was determined that the user would need to study eight major areas within IDF. These are:

A. Preparatory Operations or Activities
B. IDF Lesson Options
C. General Guidelines of Sections
D. IDF Section Options
E. IDF Units
F. IDF Intralesson Branches
G. IDF Selective Branches
H. General Procedures of IDAF

Behavioral objectives were written for each of the areas and test items were developed for each objective. The module was then prepared according to the specifications established by these objectives and test items.

As with most computer languages, IDF is learned by a process of "building" on the previous information given. As a consequence and based on the behavioral objectives, the module was arranged such that the instructional strategy was linear and non-learner controlled. By using this type of instructional strategy, it was felt the "building" process would occur more efficiently than with adaptive or learner controlled strategy.

Based on the given instructional strategy and the eight major areas of study, the specific module structure was developed so that the information in area A (see previous list) was presented before the information in area B, the information in area B was presented before the information in area C, etc. After area H was presented, a post-test was administered to determine how well the student had learned and to give the student an indication of those concepts that he/she needed to review.

The testing procedure used to determine the effectiveness of learning that had taken place during the use of the module was the pre-test/post-test design. A control group was not used because it was assumed to be very unlikely that post-test performance would be significantly affected by having taken the pre-test or by incidental learning during the time between pre- and post-test and because a comparison with another mode of learning was not to be made.

In developing the test, each behavioral objective was examined to determine precisely what was to be tested. A test question was then written for that particular objective. The purpose in writing only one test question per objective was to facilitate the revision process. Thus, based on the item statistics and the objective, the test question and/or related module material could be revised. However, in two out of the forty-three objectives, two questions were written because it was felt that the objective could be tested more adequately.

The pre-test and post-test contained the same questions with one exception. The last question in the pre-test was an inquiry as to the previous computer programming experience the user may have had. If the user answered affirmatively to this question, he/she was then asked to list the computer languages in which the experience had occured. The last question on the post-test provided an opportunity for the user to make any comments, criticisms, or remarks about the module he/she desired to give.

All testing was done on the computer. One trial was given for each question in the pre-test. Since the pre-test was the same as the post-test, no feedback was given in the pre-test as to the correctness or incorrectness of answers given. The post-test had two trials on each question to allow for typing errors. However, when the results were compiled, only the first trial answer was used. The two trials for the post-test questions was primarily a device to enable the user to correct and/or learn from his/her errors. Feedback was given on each question in the post-test indicating to the student correctness or incorrectness of the answer. In the case of two incorrect answers, the correct answer was displayed to the student.

Two separate testing periods were used, an initial testing phase and a final large group field testing phase. The purpose on the initial testing was to determine problem areas in the module so revision could be accomplished before the large group field testing occured.

After the large group field testing, the scores from the pre-test group and from the post-test group were compared. A t-test for related samples was used to test for the difference between the two. Also, using the K. R. 20, the reliability of each test was computed.

Results

The reliability of the pre-test was found to be 0.88 and the reliability of the post-test was 0.89. Forty-six students completed the pre-test and, of these, thirty-one also completed the post-test.

The means, variances, and standard deviations for each test are shown in Table 1.:

Table 1.

Test Results

Pre-test (N=46)	Post-test (N=31)
$\bar{x}$ = 9.391	$\bar{x}$ = 29.581
s^2= 44.803	s^2= 63.856
s = 6.694	s = 7.991

The null hypothesis tested was that the difference between the means on the pre-test and post-test is zero. A two-tailed test with a .05 level of significance was used. The test statistic was found to be 11.837. The null hypothesis was, therefore, rejected.

There were six items in the post-test having less than 60% of the users answering the question correctly. Also, there were eight items in 60 - 69.9% range. The remaining thirty-one items had success rates of 70% or better.

The last question of the post-test, as discussed earlier, was an inquiry or invitation to the user to make comments, criticisms, or remarks. The question was not structured in any manner so as to allow for the greatest possible range of responses.

The responses obtained can be classified into four general categories:

1. Unfavorable reaction
2. Indifferent reaction
3. Indifferent to favorable reaction
4. Favorable reaction

The responses in category one were those considered to contain negative reactions to the tutorial. Category two responses were of a nature such that no criticisms, either favorable or unfavorable, were made. The basic substance of each response in this category was a suggestion or a note about a specific answer he/she felt should be checked. Also included in this category were ideas about possible revisions. Responses in category three were those that usually contained a favorable reaction to the module and offered suggestions for possible revisions. This category contained more responses than any other category. The responses in category four were those regarded as having a favorable reaction to the module. In a few of these responses, minor suggestions and/or comments about a particular question were offered, however, most of these did not contain any particularly useful ideas.

One idea that was found to be consistent throughout all categories basically related to the mechanical nature of the terminal. This was that typing in general on a teletype is a difficult experience. Many also added that their poor typing skills was a source of considerable frustration.

Revision

The data from both the specific questions and the last general question on the post-test provide the information and the basis by which the module is to be revised. It was determined from the data that revision in both the tests and the module was necessary.

If the success rate was much greater than chance for the item, in the pre-test, it was reassessed. In the post-test, if the success rate was below 60%, the item, objective, and related module material was also reviewed. Finally, if the success rate on any material in the module not related to those examined as a consequence of the above criteria was less than 85%, it was examined. Fortunately, no items were in the case of the latter.

In the pre-test, two items had a success rate much greater than the chance success rate. These two particular questions were examined and will be rewritten. In one case, the objective was considered to be faulty and thus will also be rewritten.

In the post-test, items were classified as Type I, Type II, or Type III revision. Type I revision occured when an item had an acceptable success rate (above 60%), but was mentioned by a user or users in the comments. Type II revision was based on low success rate (equal to or less than 60%). Type III revision was based on both low success rate and user comments.

In the Type I revision category, it was found that most user comments dealt with the problem of typing an answer and having it rejected by the computer. This was primarily the author's fault for not anticipating all possible answers. Thus, these questions will contain a new, larger set of answers to help rectify the problem.

The Type II revision group contained six items that needed examination. Each was found to have a unique problem, not necessarily related to mechanical difficulties, but more to actual learning difficulties. Thus, each item was examined and will be rewritten in an attempt to solve each particular problem.

There was only one item in the Type III group. This item had a very low success rate (41.7%) and many comments. It was determined from the comments that the user did not know what he/she was supposed to do. As a consequence, an instruction for the user will be included with that item so that the user may understand the correct procedure for answering the item.

Those items in the 60% - 69.9% range were considered to be of marginal success and thus should be reviewed, but not classified. There were eight items falling within this group and when the responses were examined it was discovered that one of two errors had occured. The first was the situation discussed in the Type I revision. The other was typing errors that occured when the user attempted to enter his/her answer. Thus, to increase the success rate of these items, a more effective answer-processing technique will need to be developed.

Other revision included ideas and suggestions given by the users. These responses, in most cases, dealt with the desire to have more information in certain areas of IDF or to modify the manner in which the information was presented. For example, a number of users desired to know more about EDDOC and IDFUTI. Also, some felt a greater number of diagrams and illustrations would have helped facilitate their learning.

Summary

The major instructional goal of the tutorial is that the user will be able to create a CAI program using IDF with the author's manual functioning as a reference. This goal was almost impossible to measure during the testing procedure. The measurement of the major goal came, in fact, after the actual testing was completed. Many of the users that completed the module had actual assignments in their respective classes of creating a CAI module using any language available. Upon completion of the module, a great number of these users chose IDF as the language used for their projects. The author talked with many of the users after they had begun the programming and after they had completed the modules. He received positive reaction from almost everyone about the IDF tutorial. They said that it gave considerable insight into some of the small problems and ideas for use during the programming. They also related that many aspects which they felt would not be remembered after taking the tutorial were, in fact, recalled when the actual programming process for each began. These verbal comments to the author gave an indication that the major instructional goal was achieved with relative success.

Even though the major goal's attainment was considered successful, the testing-revision cycle will not be teminated. It is hoped that the revision in the previous section will add more power and yield better results on those portions of the module needing it. Thus, all subsequent testing-revison cycles will then produce only minor problems to be eliminated.

LIST OF REFERENCES

1. Couger, J. Daniel, "Educating the Faculty About Computers: Part I." Journal of Business Education (March 1969), 249-250.
2. Couger, J. Daniel, "Educating the Faculty About Computers: Part II." Journal of Business Education (April 1969).
3. Dyer, Charles A., Preparing for Computer Assisted Instruction. Educational Technology Publications, 1972.
4. Hansen, Duncan N. and William L. Harvey, "Impact of CAI on Classroom Teachers." Educational Technology (February 1970), 46-48.
5. Hewlett-Packard Company, IDF Author's Manual. Hewlett-Packard Company, 1973.
6. Jerman, Max and Patrick Suppes. "A Workshop on Computer-Assisted Instruction in Elementary Mathematics." The Arithmetic Teacher (March 1969), 193-197.
7. McLendon, Paul, "Teaching Teachers to Develop Software: . An EDTECH Project." AEDS Monitor (March 1969).
8. Meredith, J. D., The CAI Author/Instructor. Educational Technology Publications, 1971.
9. Porter, Catherine M., "Needed: An Education Course in Educational Data Processing." Journal of Educational Data Processing (Fall 1969).
10. Rogers, James L., "Survey of CAI Courses." ACM SIGCUE (January 1973), 27-30.
11. Spencer, Donald D., "The Computer Goes to School." AEDS Journal (Winter 1972), 35-37.
12. Spencer, Donald D., "The Computer Goes to School." AEDS Journal (Fall 1972), 3-30.
13. Spero, Samuel W. "A Computer Course for Educators." Journal of Educational Technology Systems (Summer 1973), 49-60.
14. Wallace, Scott, "Bridging the Gap Between Teacher and Programmer." Computer Decisions (June 1972).